VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Využití databáze MIB pro správu sítě

The Using of Database MIB for Network Management

2013 **Michal Czyž, DiS.**

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Zadání bakalářské práce

Student: **Michal Czyž**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2601R013 Telekomunikační technika

Téma: Využití databáze MIB pro správu sítě
The Using of Database MIB for Network Management

Zásady pro vypracování:

Dnešní počítačové sítě jsou rozsáhlé a vyžadují správu. Základem pro správu počítačových sítí je protokol SNMP pomocí něhož správce sítě dokáže spravovat aktivní pvky sítě. Všechny informace jsou soustředěny do databáze MIB. Cílem bakalářské práce je provést podrobnou analýzu dat získaných v laboratoři N312.

Práce bude obsahovat:
1. Popis databáze MIB s důrazem na rozdíl v jednotlivých verzích v rozsahu 8 stran
2. Popis protokolu SNMP s důrazem na rozdílnosti mezi jednotlivými verzemi v rozsahu 8 stran
3. Analýza dat získaných v laboratoři.

Seznam doporučené odborné literatury:

DOSTÁLEK, L., et al. *Velký průvodce protokoly TCP/IP: Bezpečnost.* Computer Press, 571 s. ISBN 80-7226-849-X.
RFC 2578 Structure of Management Information Version 2 (SMIv2)
Internet Standard STD062 "Simple Network Management Protocol"

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **doc. Ing. Jaroslav Zdrálek, Ph.D.**

Datum zadání: 16.11.2012
Datum odevzdání: 07.05.2013

prof. RNDr. Vladimír Vašinek, CSc.
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Dne: 28. 4. 2013

.....
podpis studenta

Poděkování

Rád bych poděkoval doc. Ing. Jaroslavu Zdrálkovi, Ph.D., za odbornou pomoc a konzultaci při vytváření této bakalářské práce. Moje velké poděkování patří hlavně mým rodičům, neboť nebýt jejich velké podpory, nikdy bych tuto bakalářskou práci nepsal. Za to jim děkuji.

Abstrakt

Tato bakalářská práce se zaměřuje na získání měřených parametrů počítačové sítě pomocí protokolu SNMP. V první části práce je rozebrán základní princip funkčnosti protokolu SNMP a jeho jednotlivých verzí včetně historického popisu. Je rovněž uveden popis základní hierarchie MIB, přístup k jednotlivým objektům v databázi a rozdělení MIB v rámci dostupných notací. Druhá část práce se soustředí na podrobnou analýzu databáze MIB, která byla získaná pomocí protokolu SNMP z aktivních síťových zařízení v rámci chodu testovací sítě. V závěru práce jsou shrnuty naměřené výsledky a cíle, kterých bylo dosaženo.

Klíčová slova

NMS, SNMP, MIB, CISCO, OID, ASN.1, SMI

Abstract

This barchelors thesis focuses on the problems of network management and monitoring of it by the help of the SNMP protocol. The theoretical part analyzes the primary functions of the SNMP protocol and it's individual versions, including a historical description. In this chapter, the thesis contains a detailed description of basic hierarchy MIB database with access to objects and partition of MIB within the available notations. The practical part is focused on detailed analysis of MIB which was obtained during the network testing on Cisco devices at a laboratory. The end of the thesis focuses on the result summary of shortterm statistics and static parameters of network measuring.

Key words

NMS, SNMP, MIB, CISCO, OID, ASN.1, SMI

Seznam použitých symbolů

Symbol	Jednotky	Význam symbolu
Přenosová rychlost	bit/s	Objem informace přenesené za jednotku času
Čas	s	Základní jednotka SI soustavy

Seznam použitých zkratek

Zkratka	Anglický význam	Český význam
ARPANET	Advanced Research Projects Agency Network	První počítačová síť
ASCII	American Standard Code for Information Interchange	Americký standardní kód pro výměnu informací
ASN.1	Abstract Syntax Notation One	Popis datových struktur
CAM	Content Addressable Memory	Asociativní paměť
CISCO	Cisco Systems, Inc.	Americká společnost zabývající se síťovým vybavením
CLNS	Connectionless-mode Network Service	Nespojované síťové služby
CMIP	Common Management Information Protocol	Všeobecný řídící protokol
CPU	Central Processing Unit	Procesor
DDP	Datagram-Delivery Protocol	Síťový protokol AppleTalk
DES	Data Encryption Standard	Kryptografická symetrická šifra
DoS	Denial of Service	Technika útoku na sítě
FTP	File Transfer Protocol	Protokol pro přenos souborů
HMAC	Keyed-hash Message Authentication Code	Typ autentizačního kódu zprávy
ICMP	Internet Control Message Protocol	Internetový protokol
IETF	Internet Engineering Task Force	Komise, která vyvíjí a podporuje internetové standardy
IGRP	Interior Gateway Routing Protocol	Směrovací protokol
IP	Internet Protocol	Internetový protokol
IPX	Internetwork Packet eXchange	Síťový protokol používaný v systému Novell NetWare
ISO	International Organization for Standardization	Mezinárodní organizace zabývající se tvorbou norem
LAN	Local Area Network	Lokální počítačová síť
MAC	Message Authentication Code	Kryptografická funkce
MIB	Management Information Base	Informační databáze pro správu
NMS	Network Management System	Systém pro konfiguraci, správu a sledování sítě
OID	Object Identifier	Identifikátor objektu v MIB
OSI	Open Systems Interconnection	Standardizace v komunikacích

PDU	Protocol Data Unit	Informační jednotka
RAM	Random-access memory	Paměť s libovolným přístupem
RIP	Routing Information Protocol	Směrovací protokol
SGMP	Simple Gateway Monitoring Protocol	Protokol určený pro správu sítě
SMI	Structure of Management Information	Definuje hierarchický systém pojmenovaných objektů
SNMP	Simple Network Management Protocol	Protokol určený pro správu sítě
TCP	Transmission Control Protocol	Spolehlivý transportní protokol
TCP/IP	Transmission Control Protocol/Internet Protocol	Sada protokolů pro komunikaci v síti
UBSM	User – based Security Model	Bezpečnostní model
UDP	User Datagram Protocol	Nespolehlivý transportní protokol
VACM	View Access Control Model	Regulace přístupu k objektům v MIB
VLAN	Virtual Local Area Network	Virtuální logicky nezávislá síť

Obsah

1 Úvod

Současnost z pohledu komunikačních technologií by se dala charakterizovat jako doba, ve které se nezadržitelným tempem zvyšují nároky na rychlejší, kvalitnější a spolehlivější dorozumívání. S neustálým vývojem komunikačních technologií roste také rozsah a složitost datových sítí. Na základě této skutečnosti jsou kladeny větší nároky i na člověka, který se má o administraci sítí starat. V rámci provozu jakékoli počítačové sítě je jednou z klíčových oblastí mít celkový přehled o síti a řešit problémy týkající se monitorování nejrůznějších činností, jako zahazování paketů či jejich zpoždění, řešení nepředvídatelných situací a poruch. Pokud bychom tyto problémy či stavy v síti přehlíželi, nebo je dokonce ignorovali, mohlo by dojít k omezení požadované výkonnosti a kvality služeb. Z těchto důvodů je nutné využívat síťový management.

K managementu sítě a celkové optimalizaci chodu sítě je nezbytné neustále získávat informace o jejím chování. Mezi tyto informace patří zejména nejrůznější měření, statistiky, monitorování poruch a hustoty provozu. V návaznosti na diagnostiku problémů v síti je pak prioritou najít vhodné řešení, které povede k eliminaci takových problémů do budoucna.

Tato bakalářská práce je rozdělena do čtyř částí. První část rozebírá protokol SNMP (Simple Network Management Protokol) včetně jeho historie, popsání rozdílnosti mezi jeho třemi jednotlivými verzemi, architektuře a způsobu komunikace, výhodách a nevýhodách.

Druhá část popisuje databázi MIB a její jednotlivé verze, přístup k jednotlivým objektům v rámci MIB a také se zmiňuje o databázi MIB společnosti CISCO.

Třetí část se zabývá výběrem programu, který bude dohlížet na správu a monitorování sítě využívající protokol SNMP. V této části jsou srovnány vlastnosti tří vybraných programů, přičemž jeden bude použit jako hlavní program pro analýzu ve školní laboratoři. Je zmíněna také základní konfigurace vybraného programu včetně popisu vytvořené topologie sítě.

Závěr práce, tedy čtvrtá, stěžejní část přibližuje práci se základními funkcemi SNMP protokolu, příjem trapů, a to jak při normálním, tak i záměrně ovlivněném provozu v síti. Je rovněž uveden popis informací obsažených v databázi MIB formou grafů.

Cílem bakalářské práce je získat komplexní představu o protokolu SNMP a provést analýzu dat z databáze MIB, do níž se pomocí protokolu SNMP soustřeďují všechny důležité informace potřebné pro správu sítě.

2 Management sítě

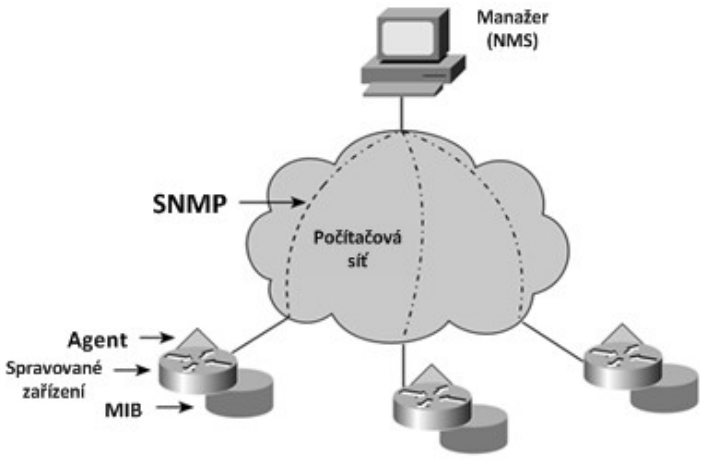
Management sítě v sobě zahrnuje prvky dohledu, kontroly, koordinace a správy otevřených systémů (komunikačních zdrojů a prostředků zpracování dat) propojených v síti, které slouží k efektivnímu usměrňování chodu sítě [1]. Co se týče podpory managementu sítě, měla by být součástí každého důležitého prvku sítě, jako je například směrovač nebo přepínač. Jen v takovém případě může být docíleno skutečně účinného managementu. Základem kvalitního managementu sítí je schopnost přenášet informace pro management [1].

2.1 Model a organizace managementu

Organizace managementu a celková vzájemná komunikace probíhá mezi dvěma základními účastníky, jimiž jsou manažer a agent (Obrázek 2.1).

Manažer neboli NMS (Network Management System) je programové vybavení, které můžeme považovat za centrální prvek zodpovídající za jednu či více činností, které pod něj spadají. Manažer svou činnost provádí pomocí agentů. Zašle agentům jednoduchý dotaz, jímž je vyzve ke sdělení určité informace. Můžeme tedy říci, že manažer má na starosti skupinu síťových zařízení, které kontroluje. Nasbírané informace shromažduje a vyhodnocuje formou tabulek, grafů a jiných různých podob, které jsou určeny administrátorovi. Na základě těchto informací je možné provést analýzu stavu, jenž nás zajímá a jenž se v síti stal.

Agent neboli klient je rovněž programové vybavení, které je nainstalováno v zařízení, z něhož hodláme data získávat. Svou neustálou činností monitoruje daný prvek a sbírá data o chodu a stavu sítě. Tyto informace se ukládají v MIB (Management Information Base). Agenti umožňují různé druhy nastavení své činnosti, například odesílání zpráv s potřebnou informací v určitých intervalech nebo sdělení zprávy manažerovi v případě poruchy či kolize. Každá takto vyslaná zpráva musí projít určitým vyhodnocením, na základě kterého je rozhodnuto, je-li zpráva důležitá, či nikoli. Tento proces je nutný k tomu, aby nedocházelo ke zbytečnému zahlcování sítě irelevantními informacemi. Je proto namístě zvážit, co je pro daný problém důležité, a na tuto věc se zaměřit.



Obrázek 2.1 Model síťového managementu
[Zdroj: http://flylib.com/books/2/59/1/html/2/images/1587050900/graphics/18fig01.gif]

Model managementu se rozděluje do několika funkčních oblastí, které můžeme na základě jejich rozdělení lépe specifikovat a tím je i snadněji zvládnout. Rozeznáváme šest základních oblastí:

- management v případě poruch,
- konfigurace,
- účtování,
- bezpečnost,
- výkonnost,
- plánování.

Rozdělení můžeme chápat jako optimální. V praxi se však vyskytují situace, kde se dá o jednoznačnosti jednotlivých oblastí pochybovat. Například, že není definována přesná hranice mezi jednotlivými oblastmi managementu, kupříkladu mezi konfigurací a výkonností. Takto by se daly vzájemně porovnat všechny uvedené oblasti a výsledek by byl stejný. Je tedy nutné uvědomit si úzkou spolupráci mezi jednotlivými funkčními oblastmi a z toho vycházet k úspěšnému dosažení stanoveného cíle [1].

Aplikace, které jsou k této práci určeny, se zaměřují na následující cíle:

- podávání zpráv v případě neobvyklé situace v zařízení,
- monitorování základních řídících informací,
- v případě manipulace se zařízeními vykonávat dohled.

3 SNMP (Simple Network Management Protocol)

Simple Network Management Protocol (dále jen SNMP) je síťový protokol, který slouží ke správě zařízení v IP sítích. Zařízení, která typicky podporují SNMP, jsou směrovače, přepínače, pracovní stanice, tiskárny, modemy a jiná zařízení [2]. SNMP je protokolem velmi jednoduchým, rozšířeným a standardizovaným. Slouží k nastavování a získávání údajů z určitého zařízení za pomoci IP adresy.

3.1 Historie SNMP

Protokol SNMP začal vznikat koncem 80. let s prvním zaznamenaným použitím v reálném provozu v roce 1988. Tento protokol vznikl na základě požadavku efektivní správy počítačových sítí.

Protokol SNMP se vyvinul jako jedna varianta protokolu SGMP (Simple Gateway Monitoring Protocol), který byl navržen koncem roku 1987 právě pro výměnu informací mezi směrovači a bránami v akademické síti [3]. Protokol SGMP umožňuje provádět nastavení a získávat informace z několika málo typů těchto zařízení. Díky aplikaci SGMP může být monitorován například stav rozhraní (zapnuto, vypnuto) nebo směrovací protokoly (RIP, IGRP, atd.) [4]. Jako druhá varianta, která byla odvozena z protokolu SGMP za pomoci organizace ISO, je protokol CMIP (Common Management Information Protocol). CMIP také zajišťuje slušnou úroveň zabezpečení (podpora autorizace, kontrola přístupu a stahování bezpečnostních logů) [5].

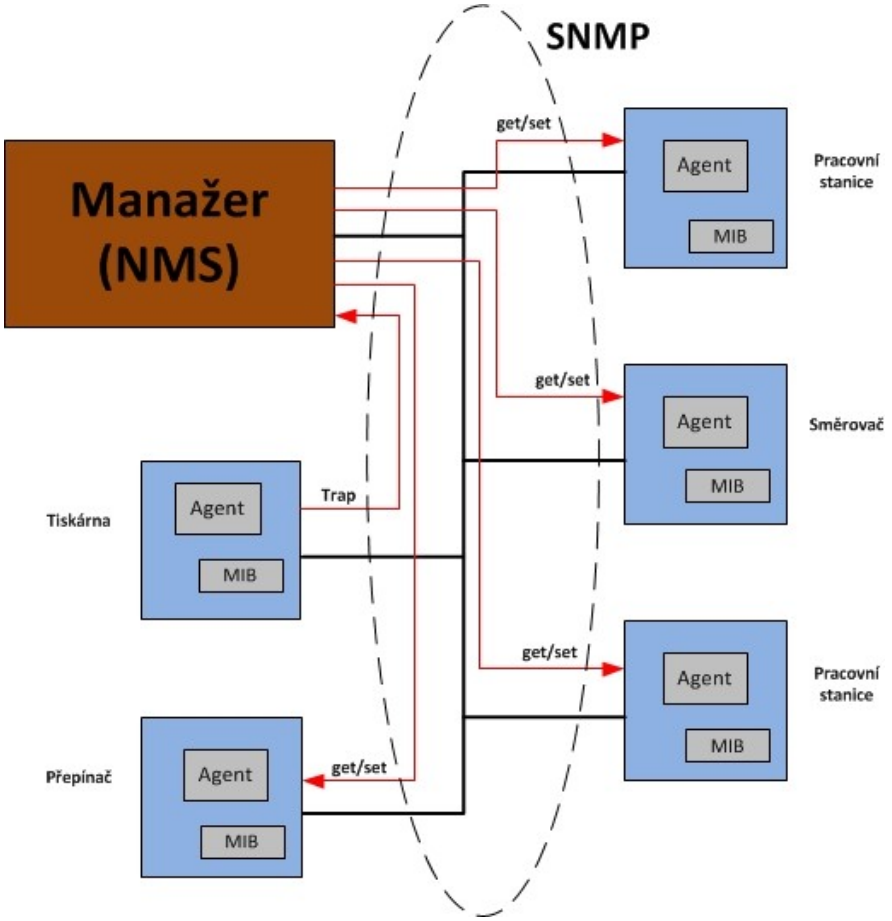
Hlavní myšlenkou byla snaha udržet a pracovat na vývoji obou verzí protokolů současně. Brzy však došlo ke zjištění, že tato cesta vede ke značné nepraktičnosti, a proto vývoj obou protokolů probíhal odděleně a hlavně nezávisle jeden na druhém. Hlavním důvodem rozhodnutí o separaci dvou téměř významově stejných protokolů byl především rozdíl mezi jejich objektovou orientací. CMIP byl na rozdíl od SNMP objektově orientován. Snahou ISO bylo vytvořit CMIP jako standard, který bude maximálně podporovat služby a protokoly s definovanou databázovou strukturou, především pro přenos pomocí protokolu TCP/IP. Za nerozšířením protokolu CMIP stál nezájem ze strany výrobců a uživatelů, stejně jako tomu bylo v případě sady protokolů OSI. Tento protokol však rychle zanikl a nedočkal se tak očekávaného rozšíření.

Vývoj SNMP probíhá již řadu let a protokol prošel několika fázemi a verzemi. SNMP je díky své všestrannosti, jednoduchosti a funkčnosti nejpoužívanějším protokolem pro management sítí [1].

3.2 Model a základní princip NMS

Model na bázi SNMP je založen na distribuci funkcí managementu mezi agenta a manažera, avšak na základě axiomu jednoduchosti, tj. požadavku minimalizace počtu a složitosti řídicích funkcí a minimálního dopadu na řízené uzly způsobeného přidáním managementu [1]. Model systému managementu se skládá ze tří základních částí. Řízený uzel jako řízený systém reprezentující síťové zařízení, dále stanice síťového managementu reprezentující pracovní stanici, která je odpovědná za sběr dat o stavu řízených objektů od řízených uzlů, a protokol managementu sítě (SNMP) vykonávající monitorování a management uzlu prostřednictvím jednoduchých operací.

Je nutné si uvědomit, že pod pojmem SNMP si nelze představovat pouze protokol, ale kompletní systém správy sítě s dalšími standardy, které tvoří databázovou část celého modelu. Pomocí vytvořeného modelu (Obrázek 3.1) lze snáze pochopit celkovou strukturu systému.



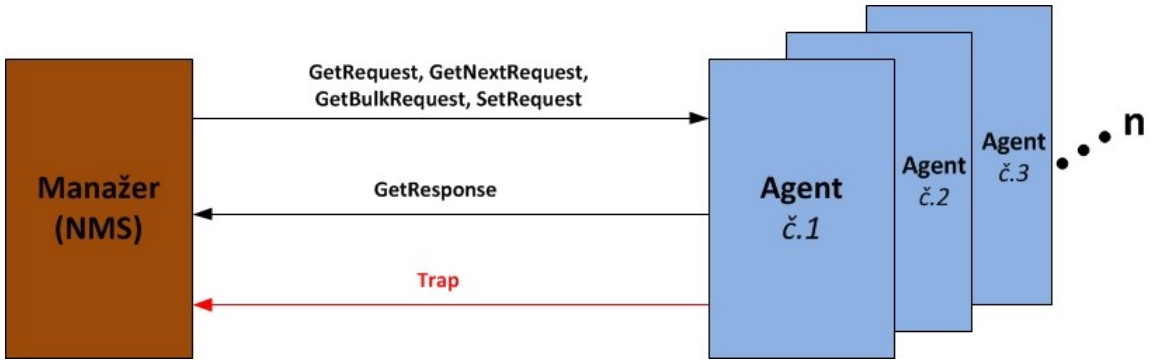
Obrázek 3.1 Struktura protokolu SNMP

Prvky, někdy nazývané elementy sítě, mohou být reprezentovány přepínači, směrovači, přístupovými body, servery, mosty, rozbočovači, tiskárnami nebo pracovními stanicemi [6].

Prvek sítě, který je určen ke správě pomocí SNMP, je považován za síťový uzel skládající se z SNMP agenta a výstupních dat sledovaného prvku. Prvky si své informace ukládají do MIB a tyto informace následně poskytují systému síťového managementu (dále jako NMS). Na základě těchto informací shromažďuje NMS údaje například o počtu uzlů v síti, o množství přenášených dat, o teplotě procesoru nebo o verzích používaných ovladačů. NMS poskytuje aplikace určené k monitorování a kontrole spravovaných prvků a zajišťuje tak množství procesů potřebných pro síťový management. Získaná data prezentuje formou statistik nebo pomocí vizuálních grafů. Nutnou podmínkou správně fungující sítě je přítomnost jednoho či více NMS v rámci sítě [6]. Funkce agenta je obvykle realizovaná softwarově. Informace, které agent vyhodnotí jako požadované manažerem, odesílá do NMS.

Agentem odesílané zprávy mohou probíhat v určitých intervalech měnících se dle požadavku nastavení administrátorem nebo při chybě (tzv. *Trap*) či jiné situaci. SNMP je tedy protokol, který umožňuje NMS (kontrolní prvek) kontrolovat agenta (kontrolovaný prvek) za pomoci výměny SNMP zpráv [7]. NMS posílá své požadavky na agenty a čeká na odpověď. Jak dlouhá je prodleva mezi požadavkem a odpovědí, záleží na nastaveném časovém limitu NMS. Pokud čekací doba na odpověď překročí určený časový limit a manažer neobdrží požadovanou odpověď, znamená to, že byl paket ztracen. Na základě této skutečnosti dochází k opětovnému zaslání požadavku.

Pro výměnu zpráv SNMP se využívá komunikace založená na UDP protokolu. Vzhledem k tomu, že UDP poskytuje nespojovou, nepotvrzenou komunikaci, hodí se do komunikace mezi velké množství agentů a manažerů v síti, protože nezatěžuje síť tolik jako TCP. SNMP agent přijímá požadavky na portu 161. NMS může poslat své požadavky z jakéhokoli dostupného portu, který má k dispozici, na port agenta s číslem 161. Odpověď agenta bude odeslána zpět na zdrojový port manažera. NMS přijímá upozornění (*Trap* a informační požadavky) na portu 162.



Obrázek 3.2 SNMP komunikace pomocí příkazů

Tento způsob komunikace je výhodný v tom, že o jednotlivých prvcích v síti může získávat informace více serverů. Stačí pouze, aby každý z nich zaslal svůj požadavek (Obrázek 3.2). Manažer může získat požadovanou informaci od zařízení pomocí operace *Get*, *GetNext* a *GetBulk*. Vysílání požadavku na změnu nastavení spravovaných prvků se provádí pomocí operace *Set*. Pro odpověď manažerovi se používá *GetResponse*.

SNMP s sebou nese mnoho výhod. Mezi základní výhody, díky nimž se protokol prosadil ve správě sítí, patří například jednoduché nastavení a použití. Je relativně jednoduchý k implementaci, výrobci dnes již uvádějí své síťové výrobky včetně podpory SNMP, a také je šetrný k procesorovému využití nebo množství spotřeby diskové paměti. I SNMP protokol má své nevýhody, například skutečnost, že informace shromažďované agenty jsou distribuovány přes mnoho uzlů napříč svou cestou sítí, či problém se zastaralými informacemi, které nejsou centrálně udržovány [8].

3.3 Formát SNMP zprávy

Vlastní formát SNMP zprávy se skládá ze dvou hlavních částí: hlavičky paketu a protokolární datové jednotky zvané PDU (Protocol Data Unit). V hlavičce je uložena verze SNMP a *community string* pro zabezpečení. Jedná se o kombinaci jména a hesla, která funguje jako autentizace

přístupových práv k agentovi. Používají se dva typy hesel, a to pro zápis a pro čtení. Hesla jsou přenášena v otevřené podobě, proto se dá takováto ochrana snadno překonat. Tato vlastnost je považovaná za jeden z nejkritizovanějších aspektů SNMPv1. K jednomu agentovi mohou totiž přistupovat různí manažeři s rozdílnými právy přístupu (Read-Only / Read-Write). Tento řetězec je součástí každého paketu z důvodu lehké oddělitelnosti komunikace jednotlivých manažerů, potřebné pro odlišení [9].

Formát zprávy SNMPv2 pro komunikaci zůstal téměř totožný s podobou zprávy pro SNMPv1. Bylo provedeno pár změn upravujících délku datových typů z 32bitových hodnot na 64bitové a také došlo k rozšíření o nové typy zpráv. Novým typem zprávy se stal *GetBulk* a *Inform*.

Nejnovější generací SNMP protokolu je verze číslo 3, tedy SNMPv3. Tato poslední generace je doplněna o vyžadovanou bezpečnost protokolu. Co se týče datových typů nebo druhů zpráv, ty se nemění a zůstávají totožné s předchozími verzemi. Je nutné zmínit podstatnou inovaci, a to změnu formátu zpráv pro komunikaci [10].

Bezpečnostní model SNMPv3 se nazývá UBSM (User–based Security Model). Kontroluje se zde stáří zprávy. Je nezbytné uchovávat při autentifikaci informace o uživatelských účtech, které jsou důležité k ověřování uživatele na základě jeho uživatelského jména, hesla a kontrolního součtu [11].

3.4 Verze SNMP

Protokol SNMP existuje v současné době ve třech verzích. Jednotlivé verze se od sebe liší hlavně podporovanými funkcemi. Výrazného využití v praxi se dostalo verzi 2, která oproti verzi 1 podporuje typ zprávy *GetBulk*. Tento typ zprávy je charakteristický tím, že umožňuje v jednom paketu přenést více než jednu požadovanou hodnotu. SNMP verze 3 v sobě zavádí podporu šifrování. Srovnání vystihující rozdílnosti mezi jednotlivými verzemi SNMP můžeme vidět v tabulce 3.1.

	SNMPv1	SNMPv2	SNMPv3
Get	✓	✓	✓
GetNext	✓	✓	✓
Set	✓	✓	✓
Trap	✓	✓	✓
GetBulk		✓	✓
Inform		✓	✓
Zabezpečení			✓

Tabulka 3.1 Srovnání verzí SNMP protokolu

3.4.1 SNMP verze 1 (SNMPv1)

Tato verze je počáteční implementací protokolu SNMP, který provádí svou činnost přes protokoly UDP, IP (Internet Protocol), DDP (AppleTalk Datagram-Delivery Protocol), CLNS (OSI Connectionless Network Service) a IPX (Novell Internet Packet Exchange) [6].

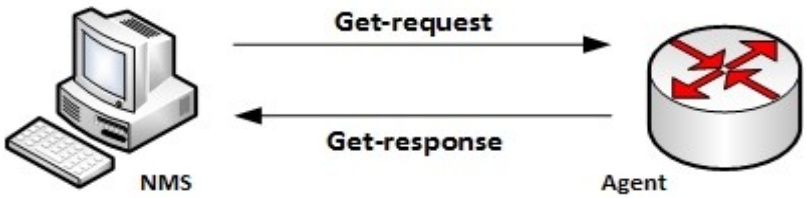
Vazba mezi agentem a NMS (nebo několika NMS), který je oprávněn provádět operace managementu na řízených objektech v rámci agenta, je charakterizována v SNMPv1 tzv. komunitou. Každá komunita se označuje jedinečným jménem. Zásadním problémem tohoto správního modelu je, že entity se sice musí autorizovat jménem své příslušné komunity, ale činí tak v otevřené formě, kdy jméno komunity lze při komunikaci mezi entitami snadno odposlechnout, protože se uvádí nezašifrované v záhlaví každé zprávy SNMP verze 1 [1].

Triviální autentizace SNMPv1 spočívá v tom, že pokud jméno komunity v požadavku neodpovídá předkonfigurovanému jménu komunity agenta, je požadavek agentem ignorován. Obvykle bývá možné nakonfigurovat agenta tak, aby akceptoval několik komunit, některé pouze pro čtení objektů, jiné i pro zápis [1].

Na základě zpětného obdržení informace od této služby protokolová entita buď zpracuje zprávu SNMP na základě profilu odpovídajícího požadované komunitě SNMP, nebo diagram v případě zjištěných chyb zničí [1]. První verze protokolu SNMP byla kritizována za velmi slabou bezpečnost. Identifikace klientů spočívá pouze v napsání hesla, které je přenášeno v jednoduché textové podobě [2]. Na základě této skutečnosti tudíž nebylo vhodné používat SNMPv1 pro vzdálené nastavování hodnot, ale spíše jen pro monitorování sítě. SNMPv1 definuje základní operace *Get, GetNext, Set a Trap.*

Operace Get:

Operace *Get* je zahajovaná SNMP manažerem (agent tuto operaci provádět nemůže, pouze odpovídá na tyto zprávy). Jedná se o základní operaci, která slouží k získání potřebných informací od agenta. Za informace považujeme data uložená v databázi MIB. Potřebná data z MIB agenta jsou označena pomocí identifikátoru objektů OID. Díky tomuto odkazu na instanci objektu OID můžeme určit přesné umístění v hierarchické tabulce i daný objekt [12]. Operace *Get* je znázorněna na obrázku 3.3.



Obrázek 3.3 Princip operace Get

Jakmile agent vyhledá požadovanou informaci, zašle manažerovi požadovaná data pomocí zprávy *Get-response*. Odpověď je identifikována stejným OID.

Operace GetNext:

Pokud nastane situace, že je požadováno získat více objektů ze spravovaného zařízení v jednom časovém okamžiku, je vhodné použít operaci *GetNext*. Může se však stát, že neznáme přesně OID (označení řádku v tabulce), a proto je nutné mít funkci, která pomůže splnit naše požadavky. *GetNext* touto funkcí disponuje a získává informace o objektech v MIB bez znalostí jejich přesných

jmen tak, že postupně prochází celou stromovou strukturu databáze MIB v lexikografickém pořadí. Pokud je dosaženo konce databáze MIB, agent zašle NMS chybovou hlášku, že bylo dosaženo konce MIB, a neexistují další objekty, které by byly zaslány s odpovědí. Není podmínkou, aby operace *GetNext* začínala vždy od kořene hierarchického systému. Lze nastavit výchozí bod, od něhož se bude dále procházet databází [1] [6] [12].

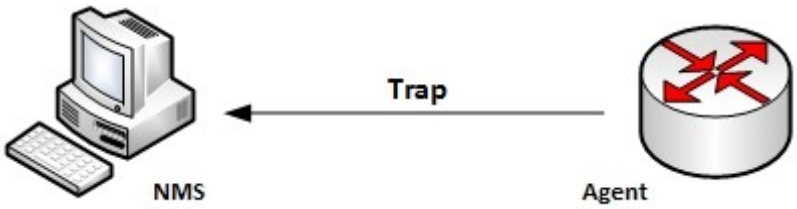
Operace Set:

Tato operace se liší od předchozích operací v jednom zásadním kroku, a tím je to, že umožňuje manažerovi měnit nastavení jednotlivých spravovaných zařízení, a ne z nich pouze data číst. Manažer je schopen změnit některé hodnoty v agentově MIB databázi nebo vytvořit nový řádek. Proměnné jsou v MIB určeny jako *Read-Write* nebo *Read-Only*. Tyto proměnné mohou být manažerem vytvořeny nebo změněny právě pomocí popisované operace *Set*. Jestliže v průběhu nastavení dojde k selhání objektu, žádné změny vyvolané touto operací nebudou uskutečněny a celá operace je zrušena. Může se například stát, že manažer bude chtít provést změny hodnoty, která má nastavený status pouze pro čtení, a v tom bude zaslána zpráva *Get-response* s oznámením o vzniklé chybě. S tímto případem se v praxi setkáváme při vzdáleném restartování zařízení administrátorem [6] [12]. Příklad operace set je znázorněn na obrázku 3.4.

Obrázek 3.4 Princip operace Set

Operace Trap:

Jediný typ příkazu vyslaný manažerovi bez předchozího vyžádání agentem jako reakce na nějakou událost, například inicializace agenta, obnovení jeho činnosti nebo jiné události specifické pro skupinu (závislé na implementaci). Zpráva *Trap* zůstává nepotvrzená, takže agent nemá jistotu, zda byla doručena [12]. Obrázek 3.5 znázorňuje operaci Trap.



Obrázek 3.5 Princip operace Trap

I přesto, že se trapy mohou během svého přenosu ztratit, neznamená to, že nejsou užitečné. Vždy je lepší, aby spravované zařízení dalo o sobě v případě nějaké poruchy vědět, než aby se o předání informace vůbec nepokusilo. Trapy mohou ohlašovat stavy síťového zařízení nebo funkčnost ventilátoru směrovače.

3.4.2 SNMP verze 2 (SNMPv2)

Navrhovaná verze 2 protokolu SNMP obohacuje výše uvedenou množinu operací SNMP verze 1 prováděných manažerem na agentovi o novou operaci *GetBulk*, umožňující manažerovi vyžádat si (k přečtení) celou množinu informací z MIB místo jediné (jako je tomu v případě *Get* i *GetNext*).

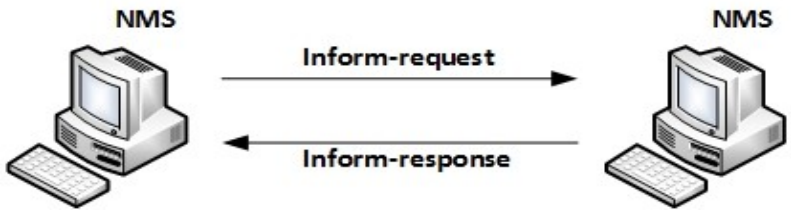
SNMPv2 také poprvé umožňuje manažerům komunikovat mezi sebou pro komplexní management celé sítě, a proto je druhou operací, neznámou ve verzi 1, *Inform* (ve formě dotazu i odpovědi) pro informování jiných manažerů o určité události.

Operace GetBulk:

Umožňuje manažerovi získat velkou část tabulky najednou tak, že operace *GetBulk* žádá SNMP agenta o zaslání co nejvíce odpovědí, ale přitom se vyvarovat chybě *tooBig*. Odpověď tak obsahuje maximální možný počet hodnot, který se do zprávy vejde.

Operace Inform:

Zpráva *Inform* je podobná zprávě *Trap*, avšak zpráva *Inform* poskytuje informační mechanizmus, na základě kterého posílá potvrzení o odeslání zprávy (Obrázek 3.6). Zpráva *Inform* může být zaslána jednak z agenta na NMS, ale také z NMS na NMS.



Obrázek 3.6 Princip operace Inform

Hlavní důraz u SNMPv2 byl kladen na zabezpečení, ale řešení, která se objevovala, nepatřila mezi šťastná řešení daného problému. Byly vytvořeny další rozšiřující verze, a to konkrétně verze SNMPv2c. Tato upravená verze obsahuje příkazy pro získávání většího množství dat pro zvýšení výkonnosti managementu aplikací, lepší chybové zprávy pro diagnostiku konfiguračních problémů a vylepšení efektivity protokolu. Chybí jí však ono nejvíce potřebné vylepšení SNMP protokolu – zvýšení bezpečnosti. Po skončení vývoje verze protokolu SNMPv2 přehodnotil IETF svůj názor a shledal implementaci SNMPv2 příliš složitou pro použití, zejména v oblasti bezpečnosti. Návrh byl proto upraven a jeho posuzování na nějakou dobu odloženo [13]. Velká nevýhoda, která pomohla SNMPv2 k čím dál tím menšímu využívání, byla vzájemná nekompatibilita se starší verzí SNMPv1.

3.4.3 SNMP verze 3 (SNMPv3)

V současné době se jedná o nejnovější verzi protokolu SNMP. Většina dnešních zařízení prakticky podporuje všechny tři verze protokolu. SNMP verze 3 doplňuje rámec managementu sítě podle SNMP verze 1 a 2 rovným formátem zprávy SNMP, bezpečnostním mechanismem zpráv a řízením přístupu k síťovým zařízením. Bezpečnostní mechanismus se zaměřuje především na ochranu zpráv SNMP při cestě sítí (před zničením, změnou dat nebo jejich odposlechem) a verifikaci zdroje zprávy SNMP. Proto se uplatňují služby zachování integrace a důvěrnosti dat, verifikace zdroje informace a dodržení časových režimů (zamezení zpoždění nebo opakování zpráv).

SNMPv3 definuje tři základní služby:

- *Autentifikace* – datový přenos od manažera k agentovi může být autentifikován, aby se zajistilo ověření identity odesílajícího.
- *Soukromí* – šifrování přenášených zpráv.
- *Přístupová práva* – agent může definovat přístupová práva, omezovat přístup manažerům pouze k některým akcím a částem dat.

SNMPv3 přichází s konceptem entity poskytující služby (*principal*), kterou může být jak jednotlivec, tak aplikace, anebo kombinace obou. Tato entita v zásadě pracuje na stanici NMS a vydává příkazy agentům SNMP. Identita *principal* vymezuje, jaká bezpečnostní opatření se na ně budou při vzájemné komunikaci vztahovat: autentizace, šifrování, řízení přístupu.

Bezpečnost je u SNMPv3 zajištěna na několika úrovních. Komunikace směrem od manažera k agentovi může být autentizována prostřednictvím MAC (*Message Authentication Code*), aby se zajistila identita a oprávněnost vysílající stanice a také integrita aktuálnosti (podle specifikace času) dané zprávy. MAC je funkcí obou zpráv, ale také tajného klíče. Hash kód zprávy (HMAC) se připojuje ke zprávě. Příjemce zprávy za použití svého klíče (sdílený tajný klíč, *secret*, musí být nakonfigurován na obou stranách mimo rámec SNMPv3) zjistí MAC přijaté zprávy a porovná výslednou hodnotu s autentizačním kódem odesílatele (MAC), který je ke zprávě připojen.

Zprávy SNMPv3 lze zašifrovat pomocí DES (*Data Encryption Standard*) pro zajištění přenášených dat, kdy komunikující strany opět sdílejí stejný tajný klíč a používají jej k zašifrování celých zpráv. Agent může využít mechanismus řízení přístupu k omezení přístupu *principal* k určitým svým datům. K tomu slouží model VACM (*View Access Control Model*), v rámci něhož může agent specifikovat, jaká data budou přístupná (jaká část MIB) anebo jaké operace s daty lze provádět [1].

4 MIB

MIB (*Management Information Base*) popisuje sadu objektů (entit), které jsou předmětem správy komunikační sítě. Za tyto objekty (například v případě přepínače) lze považovat modelové či typové označení, informace o úspěšně i chybně přenesených paketech nebo verzi firmwaru, IP a MAC adresu, záznam rychlosti daného rozhraní apod. Spravované zařízení může implementovat jednu či více MIB v závislosti na jeho funkci. Objekty, které jsou obsaženy v MIB, se nemusejí týkat pouze samotného agenta, mohou obsahovat také informace shromážděné agentem. Databáze MIB jsou velmi podobné standardním databázím v tom smyslu, že popisují jak strukturu, tak formát dat [14].

4.1 Základní hierarchie MIB

MIB jsou napsány podle pravidel SMI (*Structure of Management Information*), které stanovují strukturu databáze, tzn., jakým způsobem bude prováděna manipulace s objekty a jak se bude k těmto objektům přistupovat [15]. Problémem je však velké množství objektů, které může zařízení obsahovat. Bylo proto důležité vymyslet schéma pojmenování, aby nedošlo k vzájemné výměně objektů mezi několika různými výrobci.

Centrální registr, jenž by shromažďoval veškerá jména, by byl nepředstavitelný z důvodu nekonečné velikosti, proto byl ISO vyvinut model hierarchického stromu s názvem *SNMP Global Naming Tree*. Standardní MIB vychází z modelu, který je složen z objektů ***root, subtree*** *a* ***leaf***. Základní strukturu si lze tedy představit jako strom, jenž má ke kořenu připojeny uzly. Ty pak mohou být dále kořeny jednotlivých podstromů. Pro všechny výrobce je struktura stromu jednotná s tím, že umožňuje výrobci přidávání podstromů, které jsou specifické pro dané zařízení a daného výrobce. Myšlenka využití stromové struktury pro popis databáze MIB se osvědčila, protože zabraňovala kolizi v situaci, kdy si jednotliví výrobci definují své vlastní objekty. Pokud se každému výrobci přiřadí specifický podstrom, budou mít označené objekty jiné fyzické umístění v MIB, tedy i jinou adresu, a nemůže tak docházek ke kolizím.

4.2 Přístup k objektu v databázi MIB

Každý SNMP objekt zařízení musí být svázán s jedinečným jménem tak, aby se na něj dalo pomocí SNMP příkazů odkazovat. K přístupu k jednotlivým objektům se používají celá čísla, která jsou mezi sebou oddělena tečkou. Tato sekvence čísel se nazývá OID (*Object Identifier*).

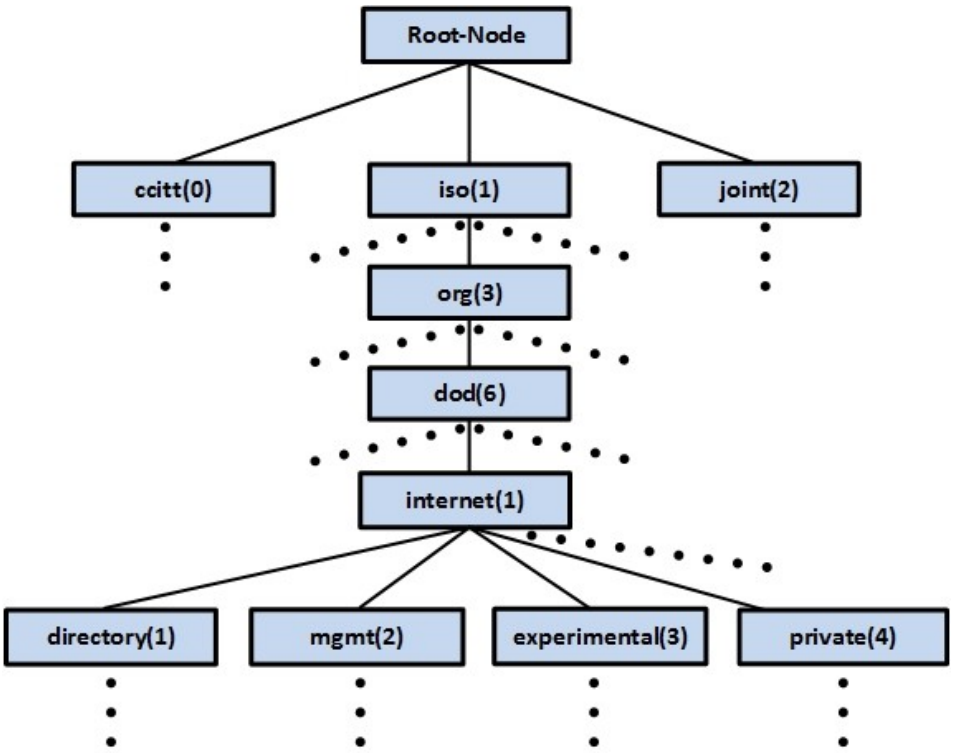
Na obrázku 4.1 je vyobrazena část základní struktury MIB. Každá část tohoto stromu se skládá ze dvou částí – popisku a číselného ***integer***. Na vrcholu stromu je kořen stromu, který se nazývá ***Root*** a dělí se na tři důležité uzly:

- CCITT – spravován organizací ITU-T
- ISO – spravován organizací ISO
- JOINT-ISO-CCITT – spravován společně ISO a ITU-T

Objekt ***iso*** se dále dělí na objekt ***org***, který reprezentuje zkratku pro konkrétní organizaci. Následuje objekt ***dod*** (*Department of Defense*) zastupující americké ministerstvo obrany. Kořeny

tohoto objektu jsou historicky spjaty s projektem ARPANET. Dalším objektem ve stromové struktuře je objekt s názvem ***internet***. Tento uzel je specifický v tom, že se dále dělí na čtyři důležité větve.

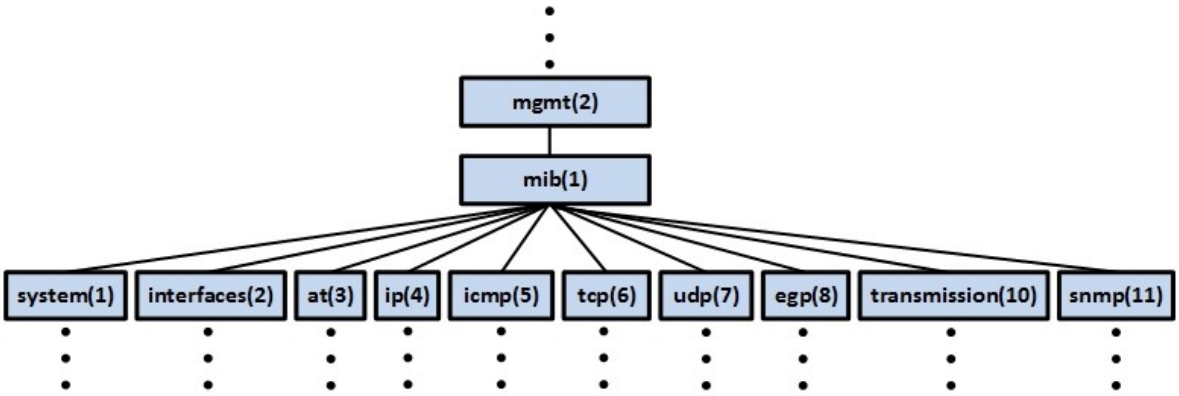
První větví je objekt ***directory*** neboli adresář, s nímž se počítá pro budoucí účely. Druhým objektem v pořadí je ***management***, který obsahuje definované objekty pro některá běžná síťová zařízení a protokoly. Tuto skupinu podporují zařízení většiny výrobců a tak umožňují jejich nezávislou správu. Objekt ***experimental*** má dle svého názvu sloužit k experimentálním účelům a zahrnuje MIB, které jsou momentálně ve vývoji. Objekt ***private***, čtvrtý v pořadí, je určen pro výrobce zařízení. Tato větev umožňuje jednotlivým výrobcům vytvářet MIB pro svá vlastní zařízení, jimž nedostačují standardní MIB. Například ***object identifier*** *.1.3.6.1.4.1.4.5* reprezentuje cestu k objektům firmy SynOptics, *.1.3.6.1.4.1.2.3* cestu k objektům Novell, *.1.3.6.1.4.1.9* cestu k objektům CISCO atd. [16].



Obrázek 4.1 Základní struktura MIB

Na obrázku 4.2 je model, který popisuje rozšířenou část databáze MIB. Konkrétně se jedná o rozšíření objektu ***management***. Důležitou a neměnnou skutečností je to, že je větev standardizovaná pro všechny a je tedy stejná v každé MIB. Tato větev se rozděluje na deset důležitých částí. První je objekt ***system***, který slouží ke zjištění stavu rozhraní, určení základního nastavení, doby běhu systému apod. Objekt ***interfaces*** je určen pro proměnné, které určují rozhraní, kde může dojít k chybě či určitému stavu, například stav vypnuto/zapnuto, počet přenesených oktetů apod. Objekty ***at*** a ***ip*** slouží k překladu adres a například zjištění verze IP, která je využívaná. Objekt ***icmp*** zobrazuje informace o odezvě nebo dostupnosti určitého zařízení, tedy chybách ICMP protokolu. Objekty ***tcp*** a ***udp*** slouží například k získání informací o využití konkrétních portů, stavu spojení či statistikách provozu. Objekt ***egp*** monitoruje směrovací protokol, díky němuž zjistíme, zda došlo k příjmu zprávy ***EGP*** s chybou nebo zda byl přijat celkový počet odeslaných zpráv ***EGP***. Pro objekt ***transmission*** nebyly momentálně

vymezeny žádné definice a poslední objekt ***snmp*** slouží k zaznamenávání výkonu a získání přehledu o počtu SNMP paketů, které byly přijaty nebo odeslány [12] [16].



Obrázek 4.2 MIB management

Pokud budeme vycházet z obrázků 4.1 a 4.2 jako ukázek základní struktury MIB databáze, můžeme určit konkrétní OID v číselném vyjádření. Jako příklad uvedu situaci, kdy potřebujeme zjistit aktuální stav rozhraní, například je-li zapnuto či vypnuto. Vyjádření, nebo také cesta, bude následující: ***.1.3.6.1.2.1.2***. Stejný dotaz, který je určen agentovi SNMP, můžeme zaslat i ve formě textového řetězce, a to v podobě: ***.iso.org.dod.internet.mgmt.mib.interfaces***. Totožným způsobem se postupuje v případě požadavku zjištění jiného OID určitého objektu.

4.3 Notace SMI

Množina pravidel způsobu definice a identifikace proměnných MIB je popsána pomocí SMI, která předepisuje použití formálního abstraktního jazyka ASN.1 (*Abstract Syntax Notation 1*), jenž je nezávislý na platformě a umožňuje výměnu dat mezi jednotlivými vrstvami síťové architektury.

Jazyk ASN.1 zavádí celosvětově jednoznačnou klasifikaci jednotlivých objektů a metody pro definice jejich vlastností v textové i číselné formě. Umožňuje definovat jednotlivé datové položky, stanovit jejich typ (tj. určit, zda jde např. o celé číslo se znaménkem, znakový řetězec či logickou hodnotu apod.), přidělit jim jméno (identifikátor). V síťových protokolech lze pomocí ASN.1 přesně specifikovat, co má počítač na mysli, když něco chce sdělit svému protějšku.

4.4 Popis SMIv1

Jak již bylo dříve uvedeno, jméno identifikuje objekt (OID). Datový typ a syntaxe je definovaná pomocí ASN.1. Tato notace je nezávislá na typu zařízení a kódování určuje, jak bude daný řízený objekt zakódován pro přenos přenosovým médiem. Podoba, do níž se data zakódují, je potřebná pro komunikaci mezi SNMP manažerem a SNMP agentem.

4.4.1 Datové typy v MIB

U první verze SMI (SMIv1) se rozlišuje typ objektu (typ proměnné), tj. definice druhu řízeného objektu, a jeho konkrétní hodnota (hodnota proměnné).

Vlastní představitel objektu daného typu je hodnota proměnné, která se skutečně využívá při monitorování a managementu. Pokud má daný typ objektu pouze jediného reprezentanta, jedná se vlastně o *skalár*, v případě vícenásobné reprezentace je typem objektu *koncepční tabulka* obsahující jednu nebo více řádek (záznamů v databázové terminologii) a v každé z nich jeden nebo více skalárních objektů označovaných jako *sloupcové objekty*. K jejich rozlišení se používá přípona ve jméně, kde *0* odpovídá případu skalárního objektu. Sloupcové objekty mají příponu tvořenou identifikátory tabulky, řádky a sloupce.

Nelze provádět SNMP operace nad tabulkami jako celkem, ale pouze nad skalárními objekty – hodnotami v tabulce. Identifikace jednotlivých polí v tabulkách se provádí pomocí indexů. Tyto indexy jsou jednoduché hodnoty nebo sady hodnot. Indexy jsou tvořeny hodnotami polí uvnitř tabulky a můžeme k nim přistupovat náhodně (příkazem *Get* a určením pozice), nebo postupně (příkazem *GetNext*) a procházet sekvenčně celou tabulku [1] [6].

4.4.2 OID – Identifikátor objektu

Každý typ objektu má své jméno a identifikátor objektu OID, který slouží k jednoznačné identifikaci objektu v rámci celé MIB. Pro snazší určení je identifikátor doplněn textovým jménem popisu objektu *(Object Descriptor*). Identifikátor je hierarchicky uspořádaná posloupnost přirozených čísel, pro větší srozumitelnost každému elementu číselného vyjádření odpovídá textové jméno (protokol používá pouze jméno v číselné formě) [1].

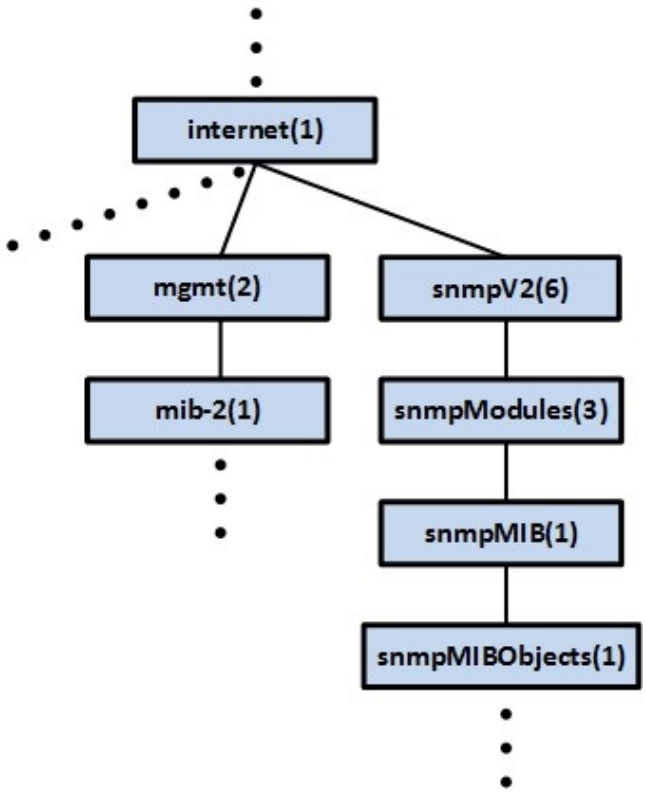
Každý OID má určité parametry, mezi nimiž je typ a odkaz na předchozí položku ve stromu. V závislosti na daném typu může být OID „listem“ stromu, tedy přímo parametrem, který je možné číst. Může být rovněž sekvencí „listů“ považovaných za tabulku nebo „větev“ stromu. Větev stromu je definována pomocí klíčového slovního spojení nebo syntaxe odvozeného složeného typu. V tomto případě se pak jedná o tabulku, která je tvořena listy. Základní shrnutí objektů v MIB a jejich OID je realizováno v tabulce 4.1.

Objekt	OID
system	*1.3.6.1.2.1.1*
interfaces	*1.3.6.1.2.1.2*
at	*1.3.6.1.2.1.3*
ip	*1.3.6.1.2.1.4*
icmp	*1.3.6.1.2.1.5*
tcp	*1.3.6.1.2.1.6*
udp	*1.3.6.1.2.1.7*
egp	*1.3.6.1.2.1.8*
transmission	*1.3.6.1.2.1.9*
snmp	*1.3.6.1.2.1.10*

Tabulka 4.1 OID jednotlivých objektů

4.5 Rozšíření na SMIv2

S nástupem protokolu SNMP verze 2 se základní notace SMIv1 (SMI verze 1) rozšířila o několik objektů na verzi SMIv2. Verze SMIv2 se rozrostla o objekt ***snmpv2*** z podstromu ***internet***. Pro lepší pochopení je na obrázku 4.3 uvedeno rozšíření verze SMIv2.



Obrázek 4.3 Rozšířený strom SMIv2

OID podstromu s více objekty má nyní cestu k objektům SNMP verze 2, například ***.1.3.6.1.6.3.1.1.*** nebo také ***iso.org.dod.internet.snmpV2.snmpModules.snmpMIB.snmpMIBObject***s. [12].

Podstrom snmpV2 v sobě nese novinky ve formě datových typů. Některé MIB jsou definovány pomocí SMIv1, zatímco jiné mohou být definovány pomocí SMIv2. Tyto dvě verze se od sebe liší způsobem svého zápisu. Například v SMIv1 je datový typ nazýván ***Counter*** a ve verzi SMIv2 je datový typ uveden jako ***Counter32***. A to platí pro všechny datové typy, které se vyskytovaly v první verzi SMI. Jedná se o změnu na ***Integer32, Counter32, Gauge32, Counter64*** atd. Pro druhou verzi SMI (SMIv2) je charakteristické, že přidává do SMIv1 specifické datové typy jako ***Bit string*** nebo ***Network addresses***. *Bit string* jsou definovány pouze v SMIv2 a určují danou hodnotu. *Network addresses* reprezentují adresy z konkrétní protokolové rodiny TCP/IP. Tabulkové indexování je mnohem jednodušší a stručnější. Došlo také ke zlepšení operací vytvoření a vymazání, které slouží k nastavení řádku v tabulce. Tato skutečnost je důležitá pro konfiguraci a kontrolu [14].

4.6 CISCO MIB

V této bakalářské práci popíši databázi MIB síťových prvků výrobce CISCO (*Cisco Systems Inc*.), protože školní laboratoř, v níž budu analýzu provádět, je vybavena přístroji společnosti CISCO. Prostřednictvím této podkapitoly bych chtěl alespoň v malé míře popsat rozšíření databáze MIB od společnosti CISCO a porovnat ji tak se standardním typem popsaným výše.

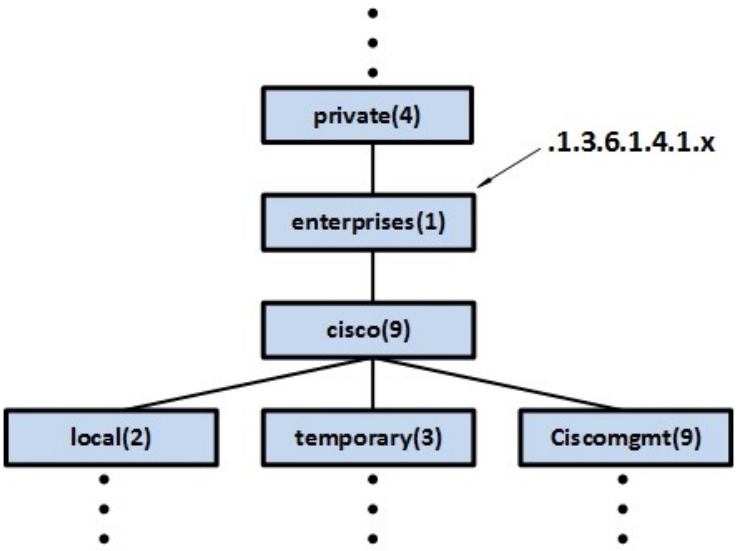
Jak jsem již v předešlém textu uvedl, kromě standardních databází MIB jsou vytvářeny také privátní databáze MIB, které si jednotliví výrobci vytvářejí na základě svých požadavků. Tyto databáze jsou rovněž implementovány do většiny dostupných síťových prvků, jako jsou například přepínače, směrovače, bezdrátové přístupové body a mnoho dalších zařízení od daného výrobce.

Privátní MIB můžeme již snadno lokalizovat na základě OID, a to pod objektem ***private*** – ***enterprise*** s číslem OID = .***1.3.6.1.4.1.x***. Symbol **x** nám konkrétněji specifikuje unikátní kód určitého výrobce. V případě společnosti CISCO je unikátní symbol potřebný pro privátní MIB roven číslu ***9***. V celém znění tedy OID = .***1.3.6.1.4.1.9 .,*** případně .***iso.org.dod.internet.private.enterprise.cisco.*** [17.]

CISCO MIB obsahuje následující podstromy:

- local (2)
- temporary (3)
- ciscoMgmt (9)

Lokální podstrom ***local*** obsahuje objekty MIB, které jsou definovány konkrétně pro CISCO IOS (*Cisco Internetwork Operating System*). Tyto objekty jsou implementovány v SMIv1. Databáze MIB, které jsou definovány v SMIv2, jsou umístěny v podstromu ***Ciscomgmt***. Databáze MIB, jež jsou definovány v podstromu ***local***, přestala společnost CISCO postupně používat a nahradila je novými objekty definovanými v podstromu ***Ciscomgmt.*** Jako příklad slouží skupina TCP, která byla dříve ve skupině ***local*** – přestala se využívat a nahradila ji skupina CISCO TCP umístěná v podstromu ***Ciscomgmt.*** [17]. Podstrom ***temporary*** slouží k experimentálnímu použití. Na obrázku 4.4 vidíme část privátního podstromu CISCO MIB databáze.



Obrázek 4.4 Privátní podstrom CISCO

5 Návrh testovací sítě

Testovací síť, kterou jsem se pro analýzu databáze MIB snažil vytvořit, musela splňovat mnoho požadavků. Jedním ze základních požadavků bylo vytvoření sítě, která bude svou topologií a volbou síťových prvků co nejvíce vyhovovat vytvořeným příkladům záměrného ovlivnění provozu v síti. Pro praktickou realizaci svého plánu jsem využil laboratoř počítačových sítí, jež se nachází na Katedře telekomunikační techniky VŠB-TUO. Předmětná laboratoř je velice dobře vybavena a určena pro praktická cvičení z různých předmětů zabývajících se sítěmi či přenosem dat. Je vybavena třemi datovými rozvaděči, v nichž se nachází mnoho aktivních prvků, které jsou ideální pro mé použití. V současné době laboratoř poskytuje dvanáct směrovačů Cisco 2800, deset přepínačů Cisco C2960, čtyři L3 přepínače Cisco 3560, dva L3 přepínače Cisco 3560X podporující 10gigabitový Ethernet. Všechna tato zařízení plně podporují protokol SNMP.

Nejdůležitějším krokem pro dostatečnou a důkladnou analýzu databáze MIB z jednotlivých síťových prvků byla volba vhodného nástroje pro monitorování sítě. V tomto případě se jednalo o program, jenž by dostatečně pokryl požadavky na přístup k SNMP zařízením s možností zobrazení hierarchického stromu MIB a jeho proměnných.

Tyto dva základní požadavky, na něž jsem se při tvorbě své práce zaměřil a které dle mého názoru nejvíce ovlivňují realizaci testovací sítě, se budu snažit popsat v následujících třech podkapitolách.

5.1 Použitý nástroj pro monitorování sítě

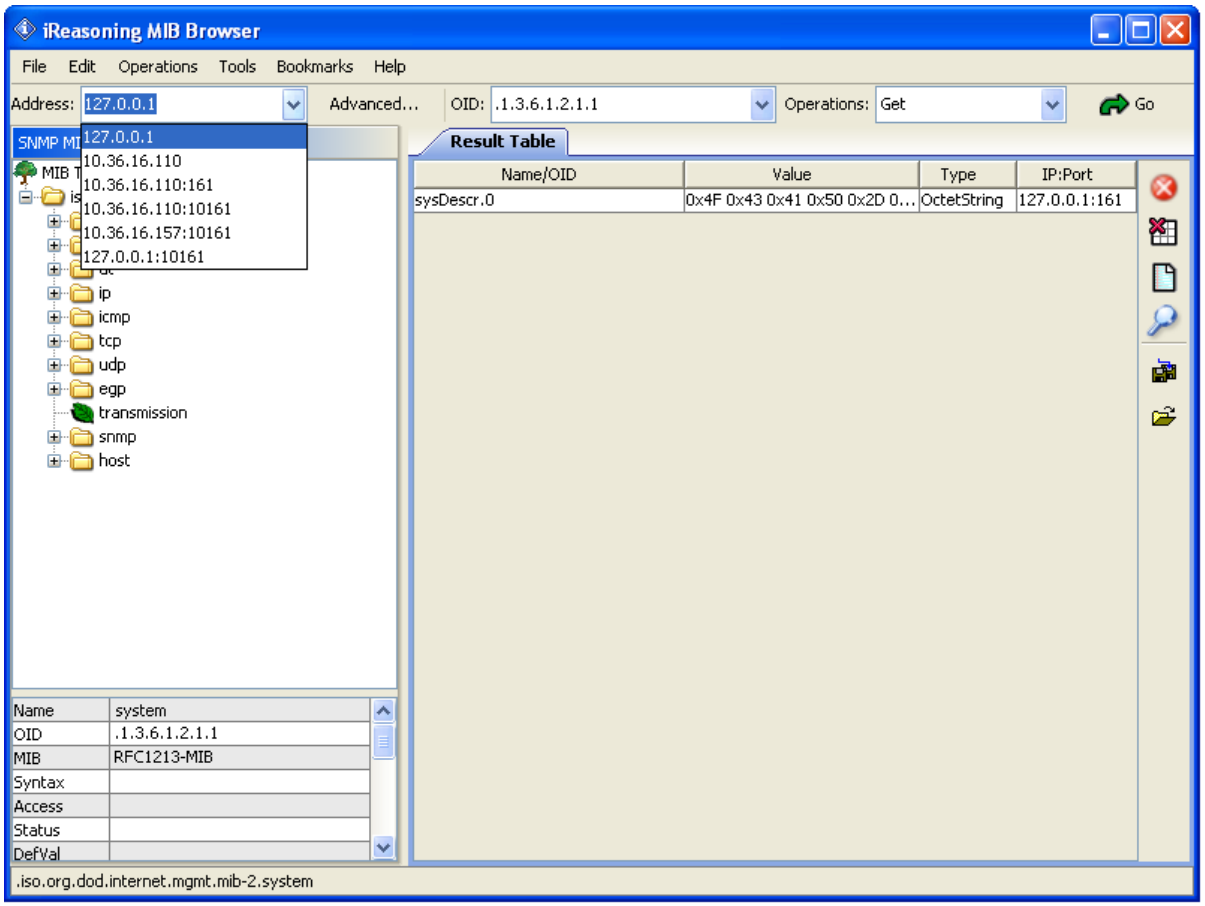
Důležitým faktorem pro úspěšný monitoring sítě je volba nástroje, v mém případě program, díky němuž budu moci přistupovat k databázi MIB jednotlivých prvků. Jelikož se jedná o bakalářskou práci, orientoval jsem se výhradně na tzv. freeware programy.

Pro sledování síťové infrastruktury existuje v dnešní době mnoho programů, které tuto činnost obstarávají. Mnohé z nich umožňují přes svá uživatelská rozhraní provádět také různá nastavení na aktivních síťových prvcích (pomocí již dříve popsané operace SNMP protokolu *Set*). Avšak důležité vlastnosti, které mají tyto programy nabízet, jsou různé statistiky či různé přehledy, a to jak v podobě textové, tak i grafické. Internet v dnešní době nabízí velké množství dostupných programů. Vybral jsem tři programy, které se budu snažit v malé míře porovnat a následovně zvolit jeden, který použiji v této práci pro své účely.

První program se nazývá *MIB Browser*. Jeho základní funkcí je náhled hierarchie SNMP MIB proměnných ve stromové struktuře. Díky programu *MIB Browser* můžeme jednoduše načíst data z MIB a nahlížet či manipulovat s daty dostupnými v SNMP agentovi. Program rovněž umožňuje převod OID z jeho numerické podoby do formátu jmen MIB. Rozhraní prohlížeče je velice jednoduché a intuitivní. Jako nevýhodu bych uvedl funkčnost programu, přizpůsobenou pouze na MS Windows.

Jako druhý v pořadí jsem vybral program s názvem *ServersCheck MIB Browser.* Umožňuje dotazování se na jakýkoli prvek, který podporuje použití SNMP protokolu. Dokáže pracovat jak s verzí protokolu SNMPv1, tak i s SNMPv2 či SNMPv3. Jakožto JAVA aplikace dovoluje pracovat na jakémkoliv operačním systému, který podporuje a má nainstalován Java Run Time.

Nejvíce mne však zaujal program *iReasoning MIB browser*. Především množství jeho různých funkcí, které mohu díky tohoto programu pro monitoring sítě využít. Náhled programů můžeme vidět na obrázku 5.1.



Obrázek 5.1 Náhled programu iReasoning MIB Browser

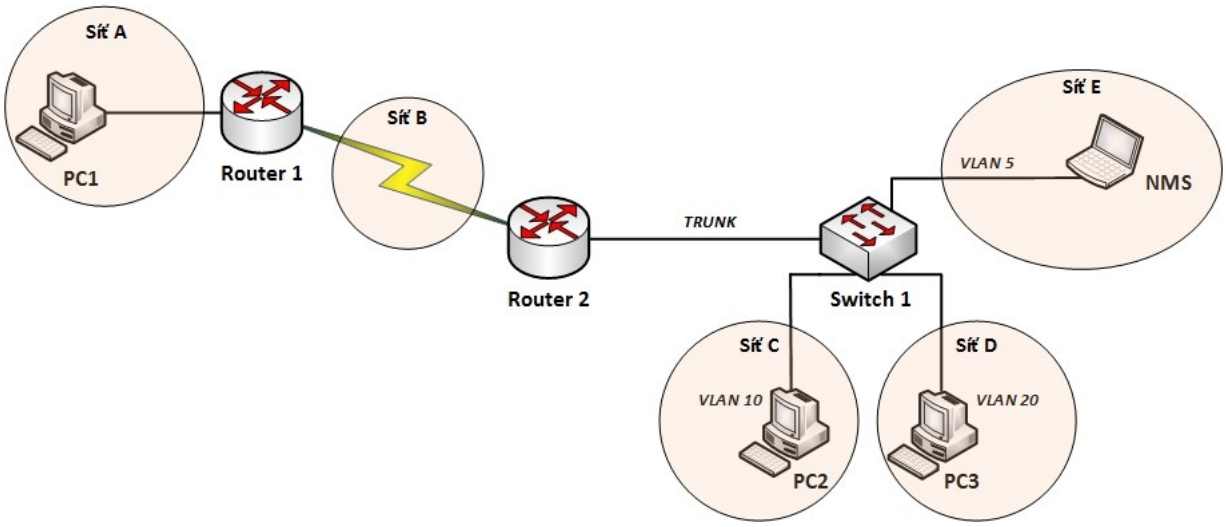
[Zdroj: https://community.cablelabs.com/wiki/display/OCORI/SNMP]

Po porovnání s většinou programů, které jsem díky internetu vyhledal a poté srovnával, se *iReasoning MIB browser* ukázal jako ideální volba pro použití v mé práci. Především mne zaujala nabízená funkce Snapshot pro zaznamenání stavu CISCO zařízení. Tato schopnost je velice užitečná pro případ, kdy chceme zaznamenat stav zařízení k danému okamžiku, a je tak určena pro rychlou obnovu dat. Snapshot je kopií dat k určitému času. Většinou je snapshot po vytvoření okamžitě k dispozici pro použití v jiných aplikacích, jako je zálohování, testování či analýza dat. Originální data jsou i nadále k dispozici aplikaci bez přerušení provozu, pouze s minimálním pozastavením datového toku. Díky snapshotům je možné rychle obnovit například omylem smazaná data bez nutnosti použití obnovy pomocí zálohovacího software. Vytváření snapshotů však s sebou nese nároky jak na výkon, tak na kapacitu datového uložiště. Možnost použití programu pro Windows, Mac OS X, Linux nebo schopnost programu interpretovat grafy pro jednotlivé parametry monitoringu považuji rovněž za velice přínosné a pro mé testování jako klíčový element.

Na základě porovnání dostupných programů určených pro práci s MIB databází jsem se tedy rozhodl použít program *iReasoning MIB browser* pro analýzu dat z databáze MIB v této bakalářské práci.

5.2 Topologie testovací sítě

V laboratoři počítačových sítí jsem vytvořil testovací LAN síť (Obrázek 5.2) se dvěma směrovači (CISCO 2800 Series), jedním přepínačem (CISCO Catalyst 2960 Series), třemi stolními počítači a jedním notebookem, který zastupuje v síti SNMP manažera (NMS).



Obrázek 5.2 Topologie testovací sítě

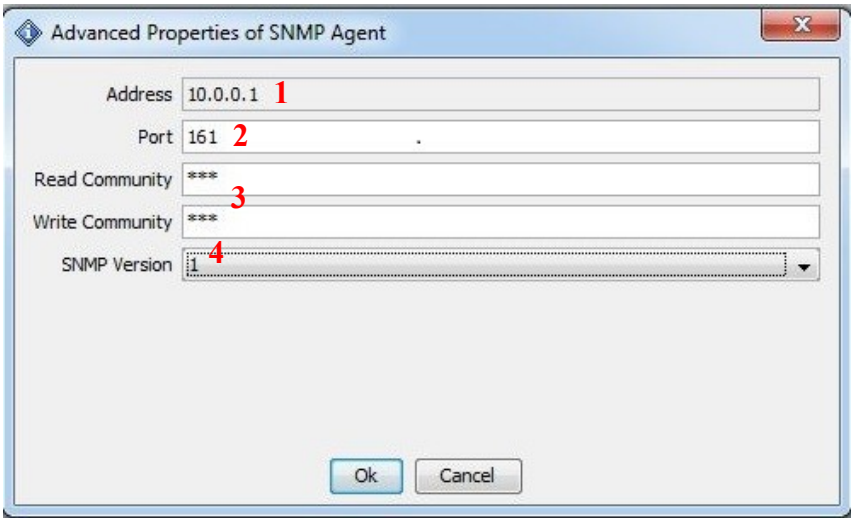
Použité rozsahy IP adres pro jednotlivé sítě a IP adresy rozhraní jednotlivých prvků jsou uvedeny v příloze B této bakalářské práce.

Testovací síť byla úspěšně nakonfigurována a konektivita propojených zařízení byla ověřena pomocí *ping* ICMP protokolu. Program iReasoning MIB browser byl nainstalován do experimentální sítě na notebook (NMS). Na všechny použité směrovače a přepínač byl nakonfigurován protokol SNMP k přijímání dotazů od manažera a zasílání zpráv trap zpět k manažerovi. Kompletní použitý postup nastavení jednotlivých síťových prvků pomocí CISCO IOS příkazů je uveden v příloze A této bakalářské práce.

Pro otestování správné SNMP komunikace bylo zapotřebí nastavit název komunity (v rámci testovací sítě byla použita jedna komunita), která byla identifikačním údajem pro použité agenty. Bylo rovněž nutné nastavit IP adresu či *hostname* NMS, který bude přijímat dotazované informace a trapy od SNMP agentů. Zvolil jsem název komunity „*vsb*“ s právem pro čtení i zápis (tedy *RW*) a IP adresu pro NMS *10.0.0.100*.

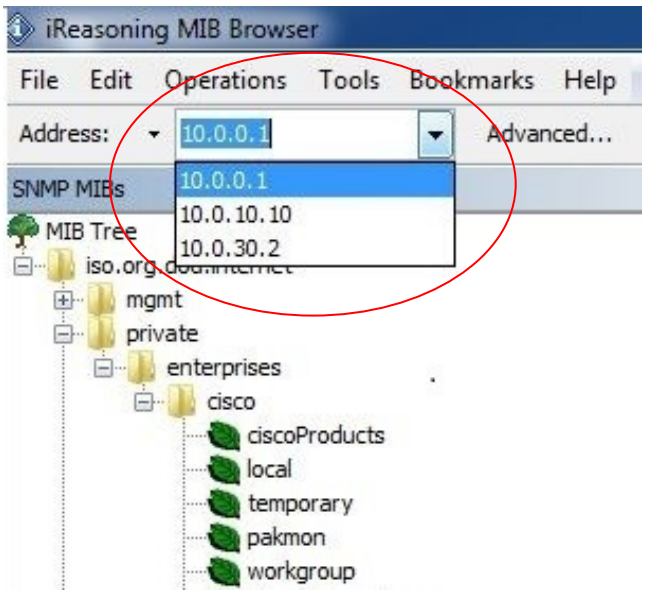
5.3 Nastavení programu iReasoning MIB Browser

Jako první je nutné vytvořit si vlastnosti agentů, kterých se bude NMS dotazovat. Příkladem zde bude obrázek 5.3 jakožto přihlášení se k SNMP agentovi *Switch1*. Tomuto přepínači byla na jednom z jeho rozhraní nakonfigurována IP adresa 10.0.0.1 (**1**), která je potřebná k tomu, aby NMS věděl, na jakou IP adresu agenta má posílat své požadavky formou dotazů. Bylo rovněž nutné uvést port (**2**), na němž bude agent komunikovat, a klíčovou informaci formou hesla (**3**) pro komunitu, v rámci níž bude komunikace probíhat. Jako poslední volba nastavení, která se uvádí, je výběr SNMP verze (**4**), a to s volbou použité verze 1, tak i verze 2 a 3.



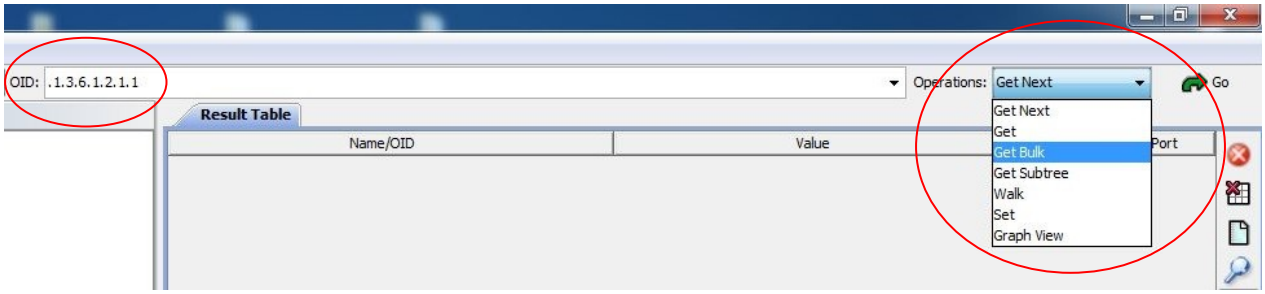
Obrázek 5.3 Nastavení SNMP agenta

Po nastavení všech údajů jednotlivých SNMP agentů můžeme pomocí nabídky adres programu iReasoning MIB Browser volit přístup k jednotlivým agentům na základě nastavené IP adresy (Obrázek 5.4).



Obrázek 5.4 Volba agenta pomocí nabídky programu

Posledním krokem k osvojení si základních manipulačních dovedností v rámci toho programu zbývá jen zmínit postup zobrazení a případné změny informací obsažených v databázi MIB každého použitého prvku sítě.



Obrázek 5.5 Volba SNMP operace na základě OID

Pomocí číselné hodnoty OID nebo i ve stromové struktuře databáze MIB, která je zobrazena v levé části plochy okna programu, můžeme dotazovat (*Get*) jednotlivé agenty za účelem získání potřebných informací nebo nastavovat (*Set*) určité hodnoty, které si přejeme u agentů pozměnit (Obrázek 5.5).

6 Analýza dat

Než jsem se rozhodl začít pracovat s informacemi obsaženými v databázi MIB síťových prvků v testovací síti, bylo nutné zvolit si verzi SNMP protokolu, s níž budu v této bakalářské práci pracovat. Jak jsem se již zmínil výše, všechny tři dostupné verze SNMP protokolu pracují téměř totožným způsobem, avšak hlavní odlišností jednotlivých verzí je způsob, jakým je přenos zabezpečen. Myslím si, že základní konfigurace SNMP verze 1 na mnou zvolené prvky sítě je naprosto účelná a vyhovující menším sítím, které jsou jednak odděleny od internetu pomocí firewallu, ale hlavně i tam, kde má administrátor přehled o veškerých činnostech a uživatelích a kde se tím pádem nevyskytuje žádné bezpečnostní riziko. Tím, že jsem zvolil SNMPv1, jsem potenciálně umožnil používání nezašifrované komunikace a navíc se vystavil riziku, že by někdo mohl přečíst *community* řetězec a vydávat se tak za SNMP manažera s možností získání přístupu k prvkům a libovolně je využívat. V této experimentální síti ovšem takové riziko nehrozí.

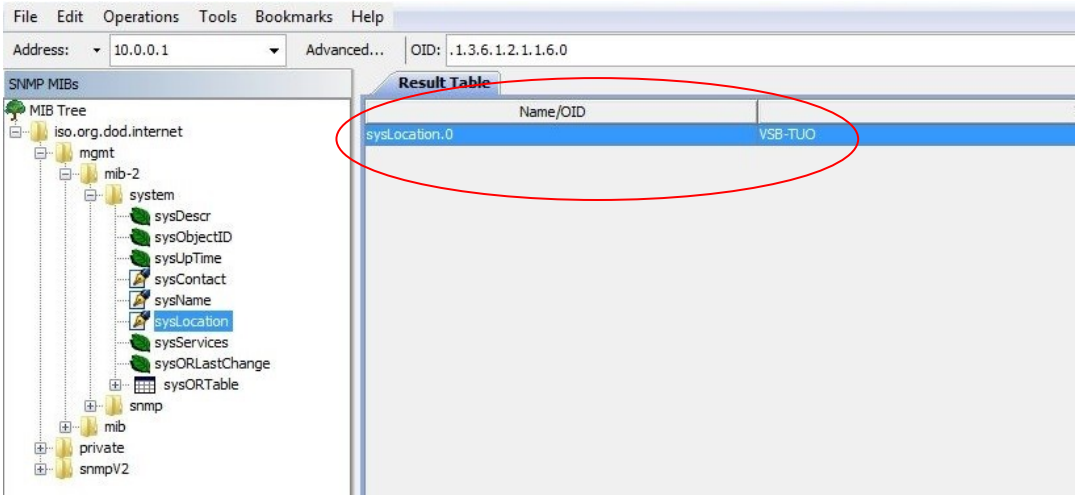
Pokud bych se však rozhodl, že svoji testovací síť připojím k internetu, je namístě zvážit riziko bezpečnosti zamezit přenosu informací do nepovolaných rukou.

6.1 Operace Get/GetBulk/Set

Zpočátku se zaměřím na zjištění informací pomocí základních operací *Get*, *GetBulk* a *Set*. Vybral jsem několik základních MIB objektů a způsob, jak s nimi zacházet a porozumět jim. Tyto objekty se mi jakožto administrátorovi jeví důležité a hlavně mi dovolují ideálním způsobem objasnit chování popisované MIB databáze.

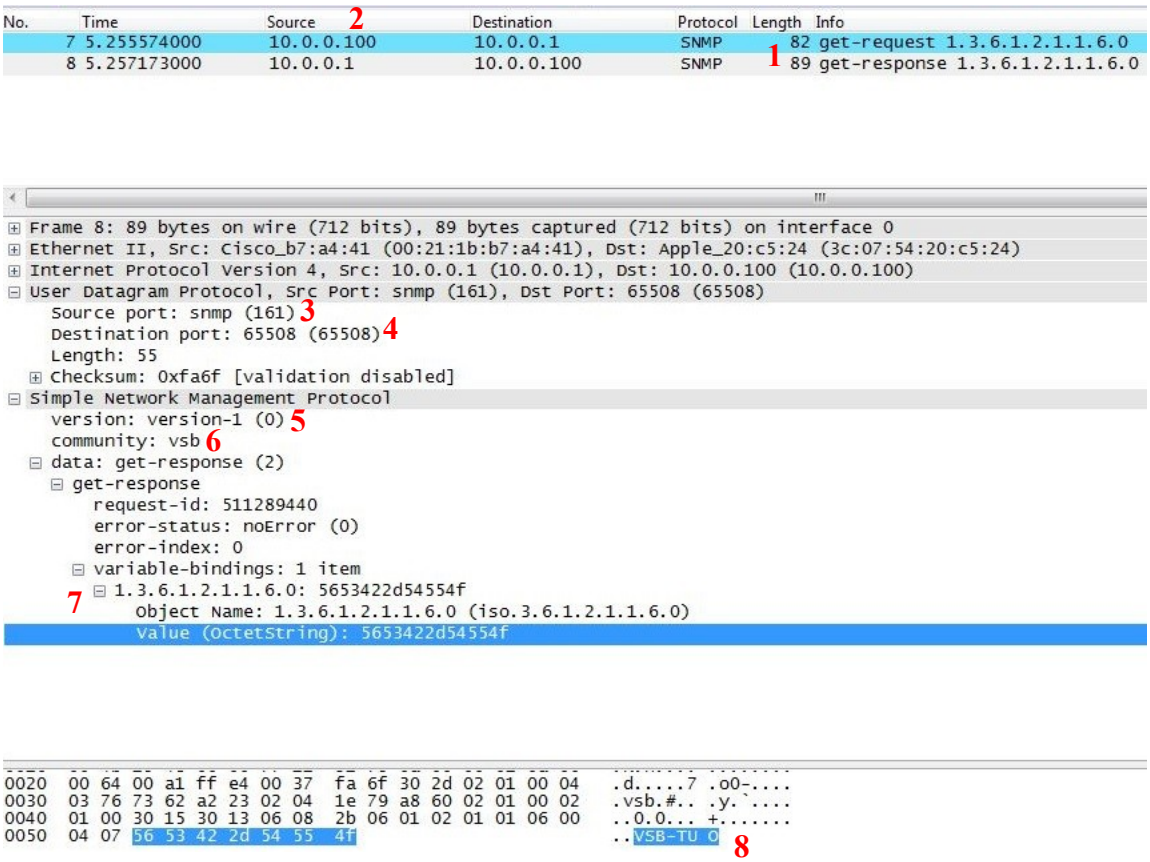
sysLocation – .1.3.6.1.2.1.1.6.0

Významem tohoto objektu je zjištění lokace, ve které se zařízení nachází (Obrázek 6.1). Jedná se o základní, i když v rámci SNMP konfigurace síťového prvku nepovinný parametr, který je velice užitečný. Příkladem může být situace, kdy mám naplánovanou údržbu na zařízení a nejsem si jistý, ve které budově v rámci areálu společnosti je tento síťový prvek umístěn. Zjistím to jednoduše pomocí tohoto MIB objektu (v mém případě budova ***VSB-TUO***).



Obrázek 6.1 sysLocation pro Switch1

Definici operace *Get* si lze ověřit například pomocí nástrojů, které slouží pro zachytávání síťové komunikace. Pro tento případ jsem zvolil paketový analyzátor Wireshark, díky němuž je možné zachytit komunikaci zařízení s nainstalovaným agentem a NMS.

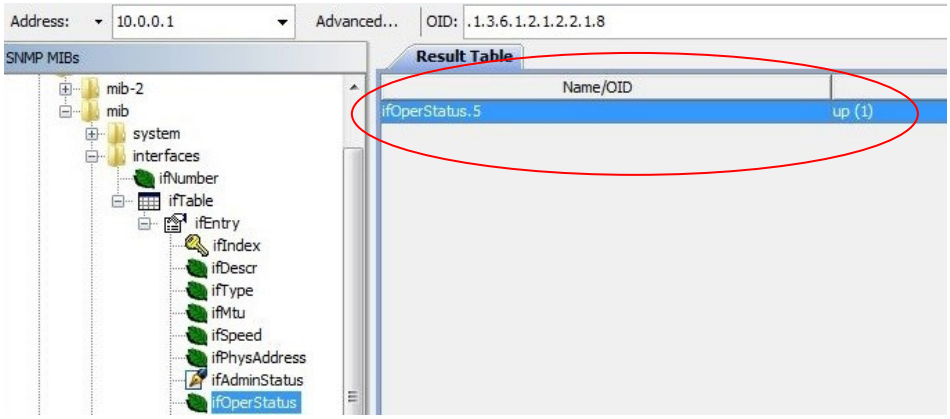


Obrázek 6.2 Struktura zprávy Get-response pro sysLocation

Na obrázku 6.2, znázorňujícím strukturu zprávy *Get-response* pro zjištění umístění prvku v síti, je možné vidět všechny důležité informace, které jsem v rámci SNMP chodu sítě nastavil. Jako základní informaci bych uvedl zobrazení dvou zpráv, a to *get-request* a *get-response* (**1**). Zpráva *get-request* byla poslána NMS se zdrojovou IP adresou 10.0.0.100 na IP adresu 10.0.0.1 (**2**), která je IP adresou SNMP agenta s označením *Switch1*. Z výpisu lze přečíst velice podstatné informace. Využití UDP s portem 161 (**3**) pro NMS a náhodně zvolený port 65508 (**4**) pro SNMP agenta. Můžeme rovněž identifikovat použitou verzi SNMP protokolu (**5**) včetně jeho nastavené komunity (**6**). Data ve zprávě *get-response* nám udávají dotazované OID objektu (**7**), ale to nejpodstatnější je odpověď *VSB-TUO* s typem proměnné *OctetString* (**8**).

Objekty týkající se rozhraní na zařízení jsou rovněž velice důležitými informacemi pro administrátory sítě. Pro příklad uvedu objekty ***ifOperStatus***, ***ifSpeed*** a ***locIfDescr***.

ifOperStatus - .1.3.6.1.2.1.2.2.1.8

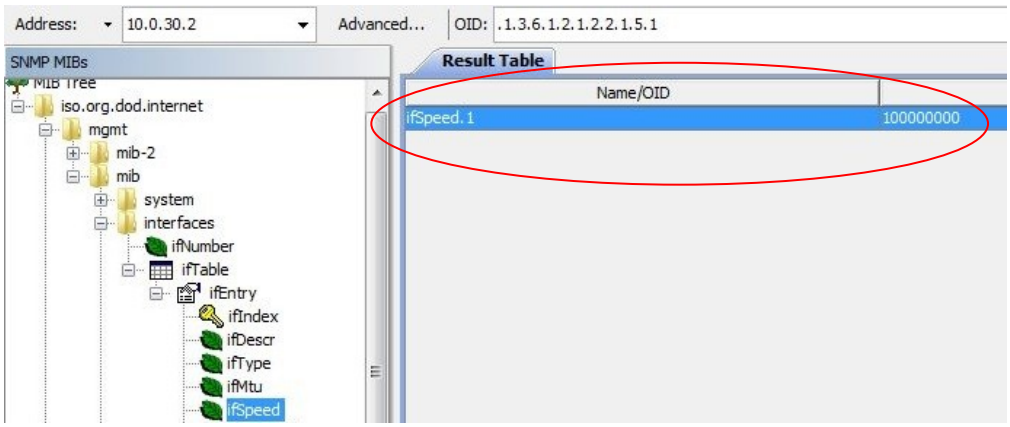


Obrázek 6.3 ifOperStatus pro Switch1

MIB objekt na obrázku 6.3 udává stav rozhraní, je-li rozhraní zapnuto nebo vypnuto. Nemusím tak například pomocí příkazu ping ověřovat dostupnost. Stačí pouze zadat patřičné OID objektu či pomocí stromové struktury databáze MIB dohledat patřičný list stromu databáze. Je důležité si uvědomit, že zadáním samotného OID *.1.3.6.1.2.1.2.2.1.8* nedocílím aktivity rozhraní. Je nutné zjistit indexy jednotlivých rozhraní (***ifIndex*** - .1.3.6.1.2.1.2.2.1.1). V mém případě jsem zkoušel ověřit funkčnost rozhraní Fa0/5 (spojení mezi *Switch1* a NMS) na *Switch1*. Na toto rozhraní byl pomocí VLAN5 připojen právě využívaný NMS (viz. Podkapitola 6.2 Topologie testovací sítě). Výsledkem dotazu je tedy **up(1)**, kde typ proměnné *integer* může nabývat tří hodnot, a to **(1)** pro zapnuto, **(2)** pro vypnuto a **(3)** pro testování. Indexování rozhraní má vždy pravidlo, že indexy do 10 000 reprezentují VLAN sítě, které jsou na zařízení nastaveny. Indexy od 10 000 výše reprezentují porty. V mém případě měl tedy OID formu *.1.3.6.1.2.1.2.2.1.8.10005*.

Pro objekty ***ifSpeed*** a ***ifDescr*** je charakteristické, že dotazováním pomocí operace *Get* zjistím například nastavenou přenosovou rychlost určitého rozhraní v bit/s nebo popis rozhraní, které uvedu proto, abych si lépe označil spojení mezi jednotlivými prvky a ulehčil si tak orientaci v rámci propojenosti topologie sítě.

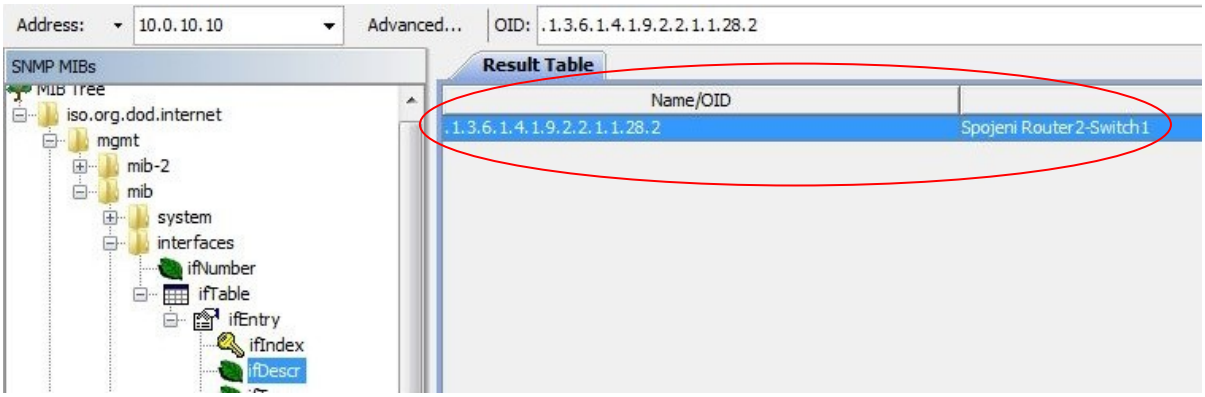
ifSpeed - .1.3.6.1.2.1.2.2.1.5.1



Obrázek 6.4 ifSpeed pro Router1

Jak jsem již uvedl, zobrazovaná výsledná hodnota pro objekt ***ifSpeed*** (Obrázek 6.4) je hodnota rychlosti rozhranní v bit/s. Na obrázku 6.4 můžeme vidět použití dotazu na popisovaný objekt pro *Router1* (s IP adresou rozhraní 10.0.30.2) s výsledkem hodnoty 100000000. Tuto hodnotu si jednoduše převedeme na hodnotu 100 MBit/s a zjistíme, že rozhraní je nakonfigurováno na rychlost odpovídající rychlosti technologie FastEthernet pro rozhraní Fa0/1 nastaveno jako spojeni TRUNK.

ifDescr - 1.3.6.1.4.1.9.2.2.1.1.28.2

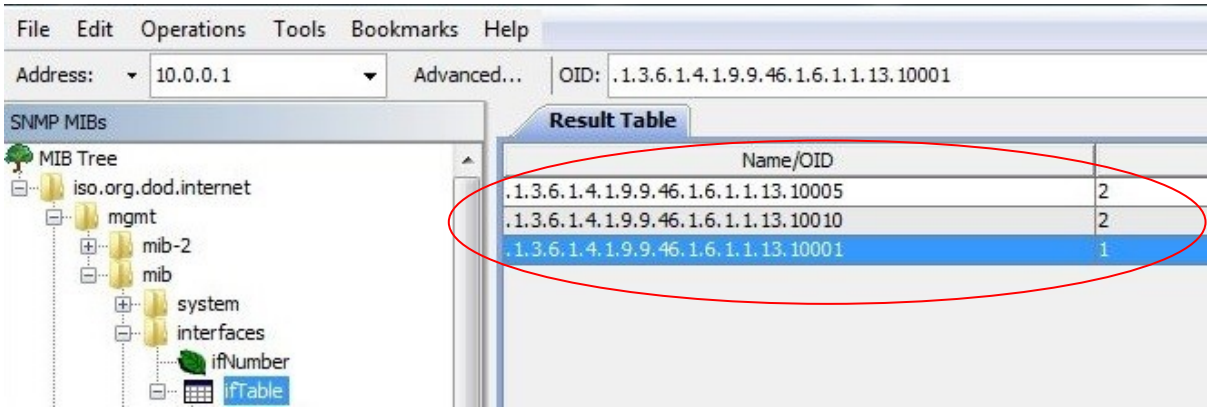


Obrázek 6.5 ifDescr pro Router2

Díky programu iReasoning MIB Browser (Obrázek 6.5) můžeme snadno zjistit výslednou hodnotu pomocí dotazu *Get* pro nastavený popis u rozhraní Fa0/1 směrovače *Router2* s IP adresou 10.0.10.10, který v tomto případě reprezentuje proměnnou *string* s hodnotou „Spojeni Router2-Switch1“.

Pokud se zaměřím na získávání informací, které pomohou administrátorovi usnadnit práci při využívání spojení MIB databáze s patřičným MIB browserem, chtěl bych v tomto případě rovněž zmínit MIB objekty, jež jsou charakteristické pro každý přepínač. Jedná se o informace týkající se nastavení VLAN. Jako příklad uvedu MIB objekt, díky kterému zjistím, je-li dané rozhraní nakonfigurováno v modu TRUNK, či nikoliv.

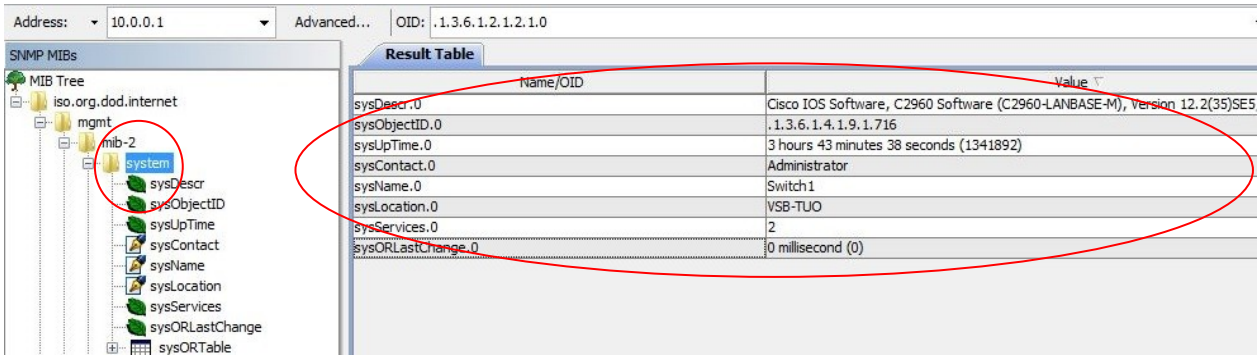
vtpTrunkPortDynamicState - .1.3.6.1.4.1.9.9.46.1.6.1.1.13



Obrázek 6.6 vtpTrunkPortDynamicState pro Switch1

V tomto případě (Obrázek 6.6) jsem postupně dotazoval SNMP agenta *Switch1* třemi dotazy *Get* proto, abych lépe zobrazil situaci, kterou chci popsat. Nechal jsem si na obrazovku vypsat tři rozhraní, a to konkrétně rozhraní Fa0/5 (spojeni s NMS), Fa0/10 (spojení s PC2) a Fa0/1 (spojení s Router2). Výsledné hodnoty proměnných MIB objektů se od sebe v jednom případě odlišují. Právě toto odlišení je pro mne klíčové, jelikož hned zjistím, že jen rozhraní Fa0/1 má hodnotu proměnné 1. Hodnota 1 popisuje stav zapnuto, zatímco hodnota 2 stav vypnuto. Výsledek je tedy ten, že rozhraní Fa0/1 pro Switch1 je opravdu nakonfigurováno dle použité topologie jako TRUNK rozhraní.

Díky operaci *GetBulk* můžu požádat SNMP agenta o zaslání většího množství zpráv najednou. Touto operací jsem se pokusil získat informace z prvku *Switch1* větve ***system*** *(Obrázek 6.7).*

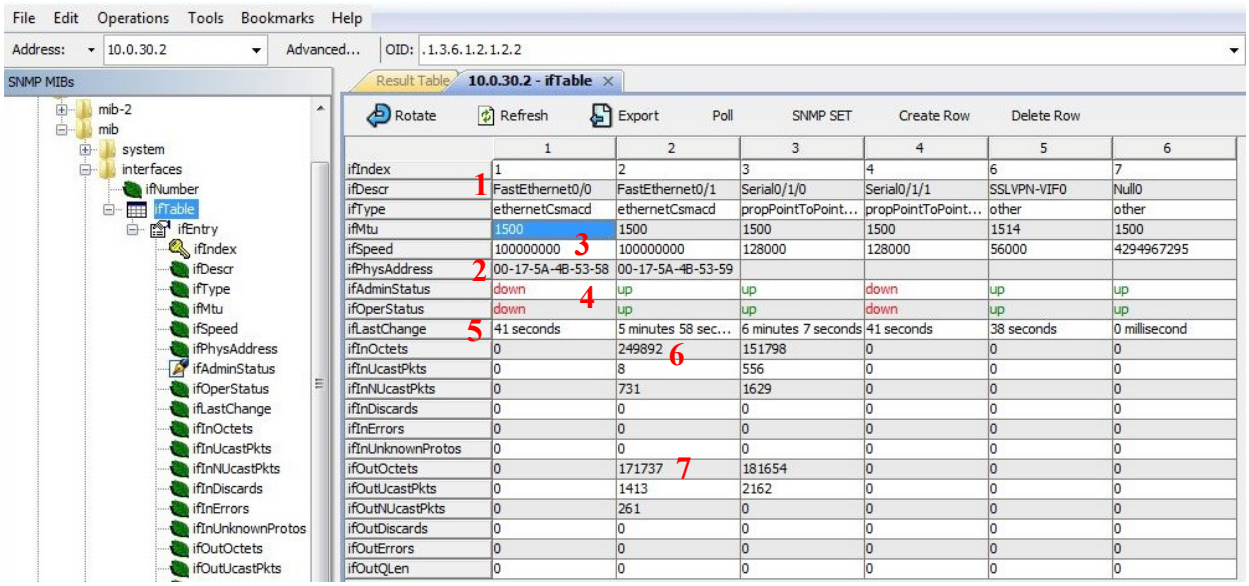


Obrázek 6.7 GetBulk pro větev system Switch1

Na základě informací obsažených v této větvi mi program zobrazil základní informace přepínače, jako je použitá verze IOS, momentální doba, po kterou je zařízení při dotazu v aktivním stavu (3.43:38), nastavené jméno kontaktní osoby (Administrator), název prvku (Switch1), lokaci prvku (VSB-TUO), vrstvu referenčního modelu OSI, na němž pracuje dotazovaný prvek (číslo 2 jakožto linková vrstva referenčního modelu OSI), či poslední dobu změny (0 ms).

Při použití programu iReasoning MIB Browser můžeme pomocí příkazu GetBulk zjistit určité MIB objekty obsažené v SNMP agentech, jako je například zobrazení informací o jednotlivých rozhraních, adresových tabulek či směrovacích tabulek u přepínačů atd. Na obrázku 6.8 můžeme vidět výpis objektu ***ifTable*** pro *Router1*. Pomocí tohoto výpisu mohu snáze určit vlastnosti jednotlivých rozhraní daného prvku. Je zde myšlen například popis rozhraní (**1**), fyzická adresa rozhraní (**2**), nastavená přenosová rychlost rozhraní (**3**) či status činnosti rozhraní (**4**) nebo čas poslední změny konfigurace (**5**), počet přijatých/odeslaných oktetů (**6**,7) a další informace.

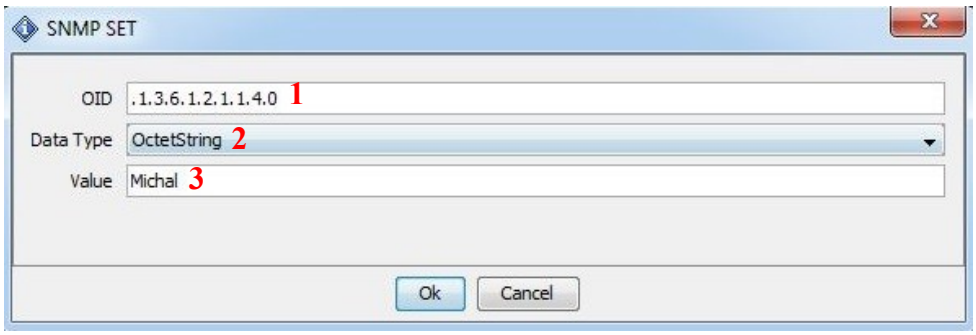
Program umožňuje exportovat aktuálně zobrazený výpis do souboru formátu Microsoft Excel. Při vyexportování tabulky a jejím následném otevření jsem dospěl k zjištění, že uložené informace jsou absolutně nečitelné a písmena neidentifikovatelně rozházená. Tuto funkci bych programu vytkl jako zcela nepoužitelnou. Soubor jsem uložil k nahlédnutí na přiložené CD.



Obrázek 6.8 GetBulk pro ifTable Router1

Pomocí příkazu *Set* mohu měnit hodnoty MIB objektů u dostupných SNMP agentů. Operaci *Set* jsem vyzkoušel například na změnu kontaktu pro *Switch1* pomocí objektu ***sysContact***.

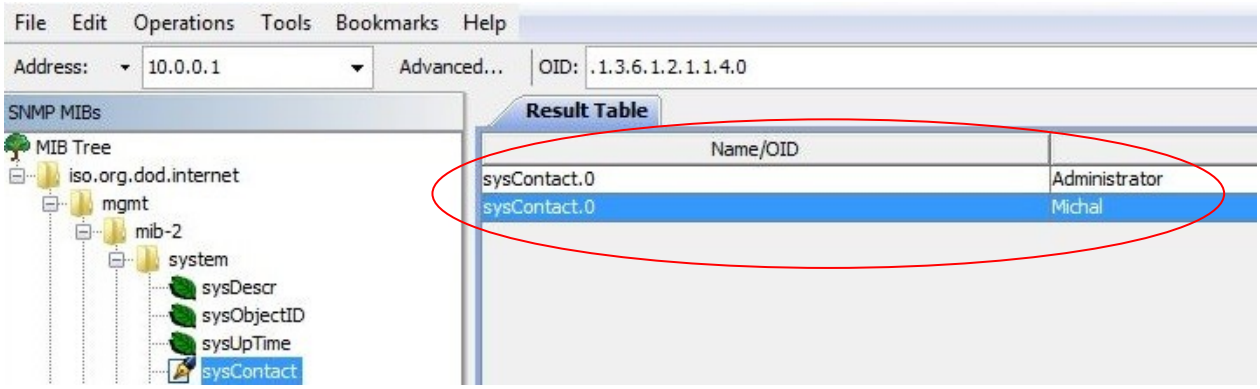
sysContact - .1.3.6.1.2.1.1.4.0



Obrázek 6.9 Set pro Switch1

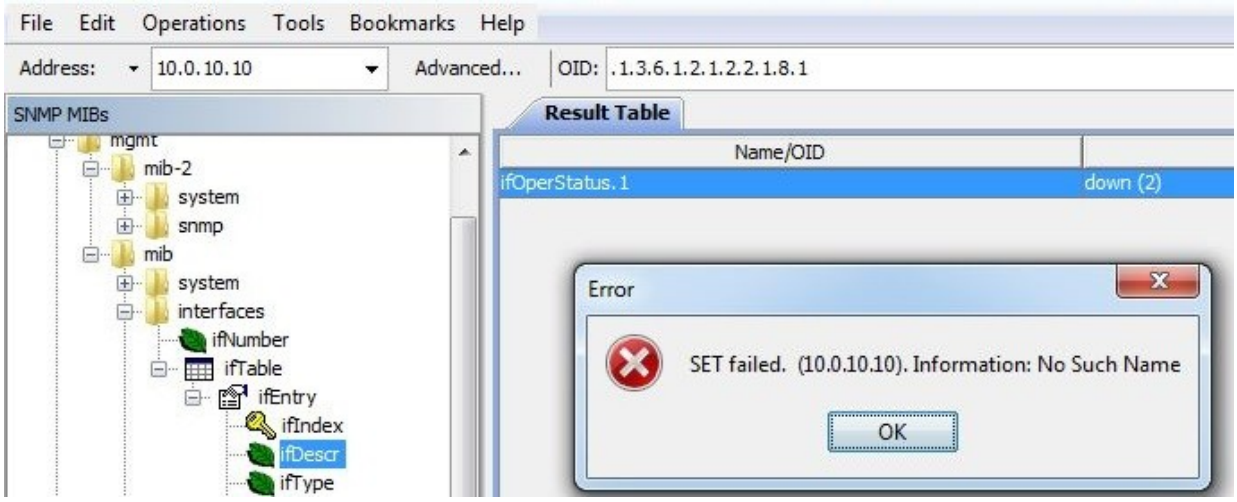
Záležitost změny nastavení síťového prvku *Switch1* operace *Set* je pomocí programu iReasoning MIB Browser jednoduchá záležitost. Jak můžeme vidět na obrázku 6.9, vše, co potřebuji je číslo OID (**1**), které je cestou k objektu ***sysContact*** ve stromové struktuře databáze MIB, datový typ (**2**), který hodlám změnit, a nakonec hodnotu (**3**), kterou si přeji uložit.

Na obrázku 6.10 jsem pomocí příkazu *Get* na objekt ***sysContact*** popisovaného přepínače zobrazil původně uložené jméno (v mém případě **Administrator**). Po provedení operace *Set* jsem opět pomocí příkazu *Get* zobrazil požadovanou informaci, avšak již se změněnou hodnotou na **Michal**.



Obrázek 6.10 Změna sysContact pomocí operace Set

Vyzkoušel jsem operaci *Set* na MIB objekt ***ifOperStatus***, který jsem již popisoval výše. Napadlo mne, že jako administrátor bych ocenil možnost měnit stav rozhraní (zapnuto/vypnuto) právě pomocí testované operace *Set*. Dle mého zjištění jsem však narazil na problém, který spočíval v nemožnosti měnit hodnotu objektu ***ifOperStatus.*** Po vyhledání informací týkajících se zmiňovaného objektu na internetu jsem zjistil, že tento objekt má implicitně a neměně povolen přístup pouze pro čtení. Během požadavku na změnu hodnoty se objevila následující výstraha (Obrázek 6.11).



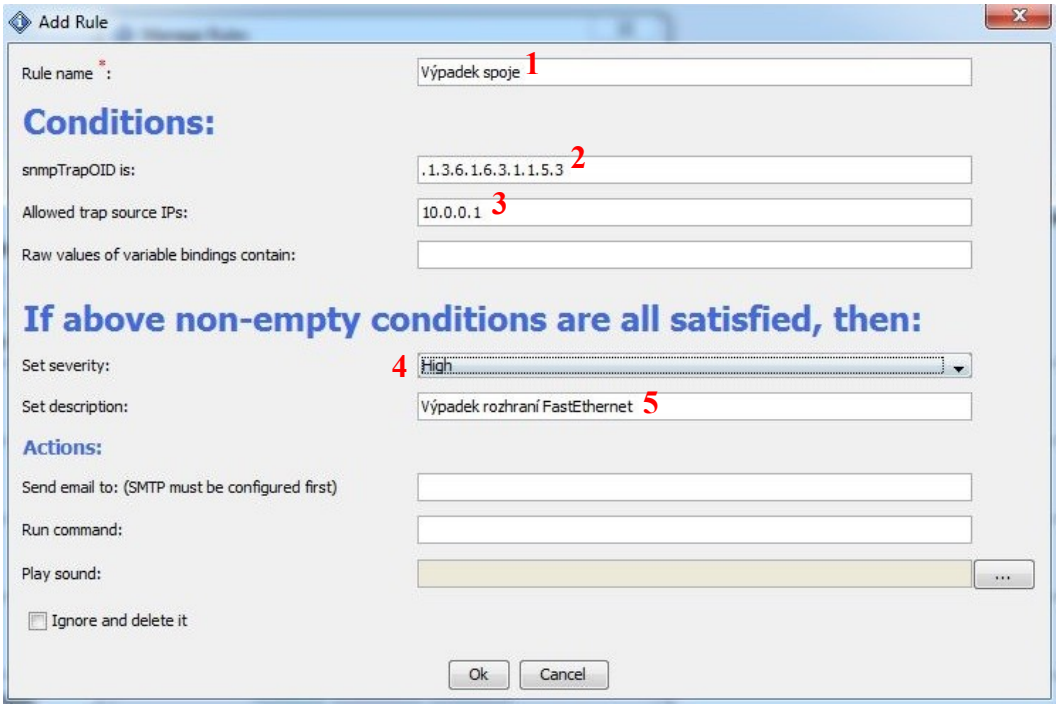
Obrázek 6.11 Chybová informace pro Set

Jako administrátor, jenž používá tento program za účelem správy sítě pomocí SNMP protokolu, bych ocenil, aby program při jakémkoli špatně zadaném příkazu na změnu hodnoty MIB objektu vygeneroval informaci, že se jedná o změnu objektu, který má povolen přístup jen pro čtení. Jedná se přece o pravidla přístupu, která se nemění a budou tedy stále stejná. Aktuálně vygenerovaná informace programu mi neposkytne žádnou směrodatnou informaci, která by mi usnadnila rychlejší zjištění důvodu, proč tuto operaci nelze provést.

6.2 Trapy

Díky SNMP protokolu může NMS přijímat od svých SNMP agentů stavové informace nazývané *Trap*. Tyto informace jsou velice užitečnou pomůckou pro administrátora, protože mu umožňují získat údaje o aktuálním dění v síti, a to jak čistě informativního charakteru, tak i výstražné údaje upozorňující například na poruchu funkčnosti linky či narušení bezpečnosti. Během testování jsem simuloval čtyři typy trapů, které bych chtěl popsat a ukázat jejich chování v reálném provozu.

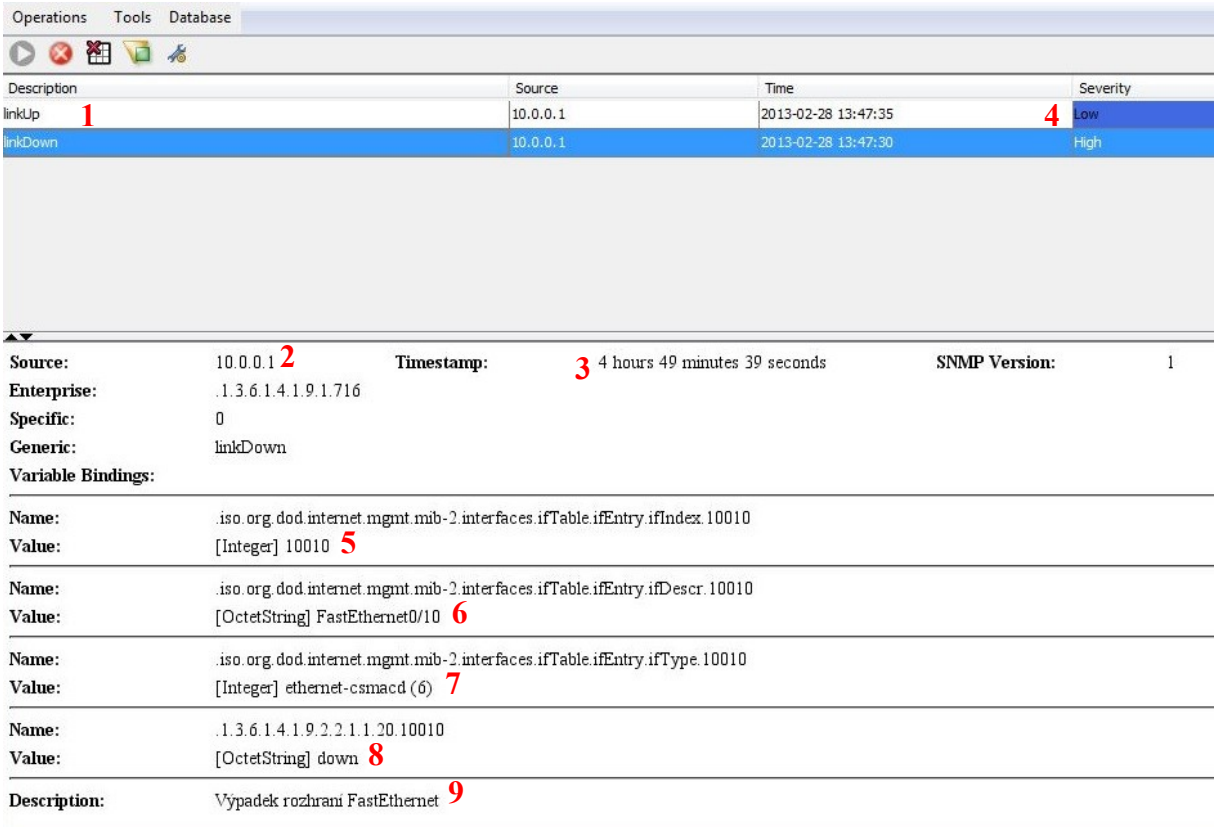
Než začnu popisovat uměle vytvořené situace narušení sítě, je nutné vytvořit v programu iReasoning MIB Browser pravidla pro jednotlivé typy trapů. Tato záležitost není sice povinná, ale umožňuje lepší orientaci a vyhodnocení situace administrátorem. Náhled tabulky pro vytvoření pravidla můžeme vidět na obrázku 6.12.



Obrázek 6.12 Nastavení pravidel pro trapy

Jediné, co je nutné uvést je jméno (1), díky kterému se budeme v seznamu vytvořených pravidel lépe orientovat, dále pak OID (2) trapu a IP adresu (3) NMS. Pro administrátora je velice užitečnou funkcí nastavení barevného označení s prioritou (4) trapu a popisem události (5) umožňující lepší orientaci.

Jako první popíšu *Trap*, který obdržím, když v síti dojde k výpadku spojení mezi dvěma zařízeními. Důvodem může být například fyzické poškození kabelu či jeho přepojení do jiného rozhraní, které již nemusí odpovídat nastavení pro danou VLAN v případě přepínače. Při vzniku této situace je nutné problém lokalizovat a následně jej řešit. Díky přijatému trapu na NMS snadno zjistím, které rozhraní a na jakém síťovém prvku přešlo do stavu vypnuto. V případě použití mé topologie se bude jednat o výpadek spojení (rozpojení kabelu) na přepínači s označením *Switch1,* konkrétně pak rozhraní Fa0/10, na němž je připojeno v rámci VLAN10 právě PC2.



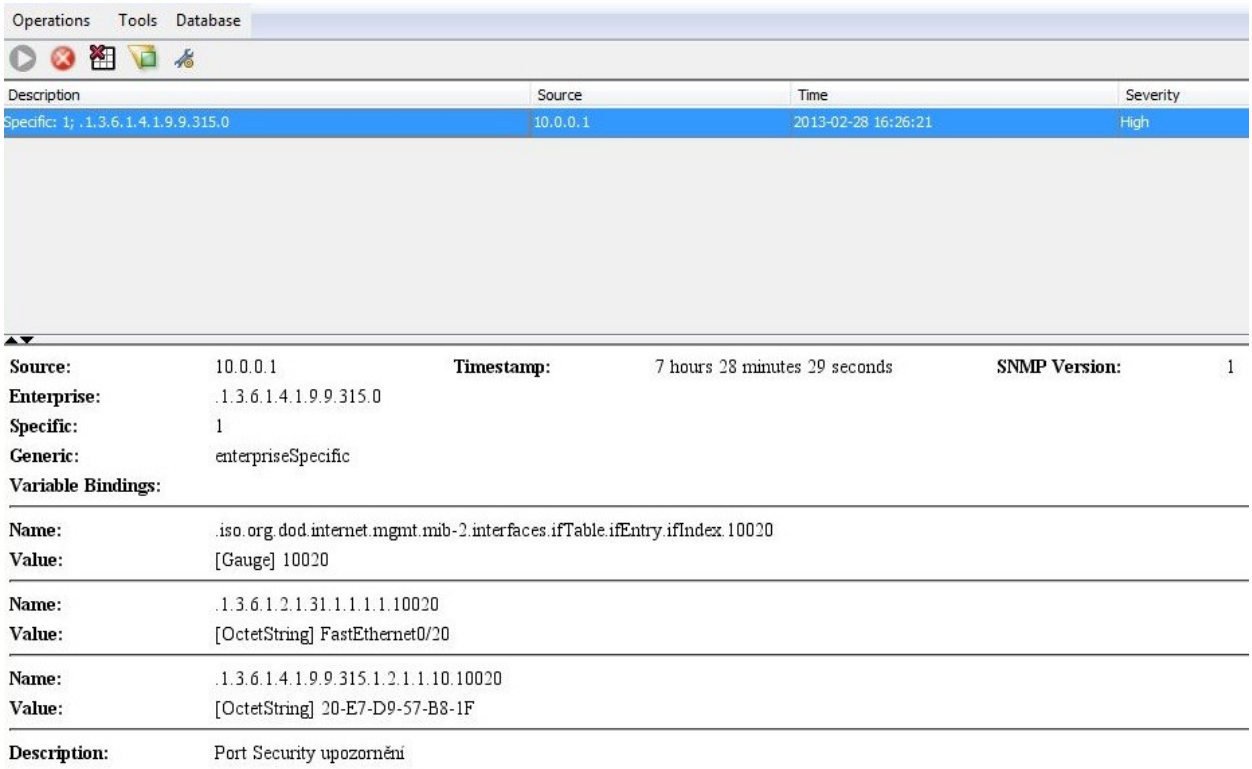
Obrázek 6.13 Trap výpadek spojení

Během několika málo sekund se v logu událostí objeví informace o poruše. Na obrázku 6.13 je uvedeno, jakou informaci jsem od svého agenta obdržel formou SNMP trapu. V horní části okna programu je vidět popsaná informace (1) (linkUp/linkDown) s IP adresou agenta (2), a to včetně zobrazení času (3), kdy k dané události došlo, a jakou má daný *Trap* nastavenou barevnou prioritu (4). V těle zprávy pak snadno identifikuji závadu. Mohu vidět, že se jedná o prvek ***Switch1*** s indexem rozhraní ***10010*** (5) a označením ***FastEthernet0/10*** (6). Je rovněž uveden typ rozhraní (7) a nejdůležitější informace o aktuálním stavu rozhraní (8), tedy ***down***. Tabulka pravidel pro jednotlivé trapy umožňuje uvést také popisek vzniklé události. Zvolil jsem větu „Výpadek rozhraní FastEthernet“ (9). V případě obnovení spojení dojde k zaslání trapu s informací o změně stavu rozhraní na stav zapnuto. Zpráva bude svým obsahem téměř totožná s výše popsanou zprávou – až na malé změny týkající se například popisku, že došlo k navázání spojení. Díky této trap zprávy již snadno a rychle lokalizuji vzniklý problém, můžu jej v relativně krátkém čase odstranit a uvést tak síť zpět do její funkční podoby.

Druhou simulací trapu, kterou jsem v rámci testovací sítě vytvořil, je útok na přepínač *Switch1* pomocí MAC address flooding neboli DoS (Denial Of Services) útok. Tento typ útoku jsem si vybral, protože se jedná o nejběžnější typ útoku na přepínač. MAC address flooding se dá popsat jako útok, který je založen na zaplavení přepínače uměle vytvořenými MAC adresami. Cílem potencionálního útočníka je vyčerpat paměť přepínače určenou pro ukládání tabulky MAC adres (CAM tabulka) tak, aby pomocí zasílání velkého množství rámců s neplatnými zdrojovými MAC adresami zcela tuto paměť přeplnil. V situaci, kdy je CAM tabulka naplněna, přepínač nevytváří nové záznamy. Na základě této skutečnosti vyhodnotí napadený přepínač situaci tím způsobem, že unicastova

komunikace určená pro cílovou MAC adresu je zaslána na všechna rozhraní mimo rozhraní příchozího. Tento útok má za následek, že útočník jednak dostává informace z celkového provozu na přepínači, ale hlavně se zvýší celkový provoz v síti a výsledkem může být příliš velké vytížení napadeného přepínače. Tento stav jsem se pomocí programu iReasoning MIB Browser snažil zachytit jak pomocí SNMP trapu, tak pomocí vykreslování grafů pro aktuálně vytíženou paměť typu RAM a CPU monitorovaného SNMP agenta. Část týkající se grafů pro vytíženou paměť RAM a CPU přepínače je popsána v podkapitole 6.3 Grafy zatížení použitých prvků.

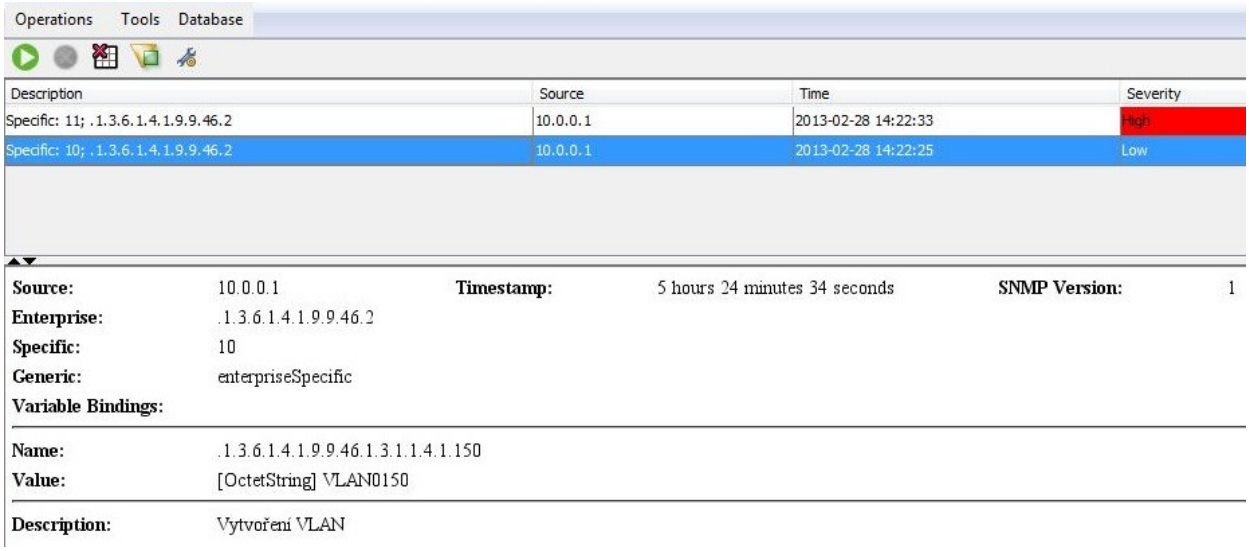
Pro vygenerování velkého množství MAC adres jsem použil program **DSNIFF**, který byl nainstalován a poté spuštěn na PC2 v rámci použité topologie sítě. Než jsem pomocí programu **DSNIFF** spustil generování uměle vytvořených MAC adres, bylo nutné nakonfigurovat na rozhraní Fa0/10 (spojení s PC2) přepínače *Switch1* takzvanou *Security port*. Tato funkce umožňuje blokovat rozhraní přepínače, pokud se o přístup na dané rozhraní pokusí jiná než povolená MAC adresa nebo pokud je překročeno množství MAC adres, které administrátor dle svého uvážení zvolí. Pro názornou ukázku jsem zadal hodnotu 10 jakožto maximální počet MAC adres povolených na testovaném rozhraní.



Obrázek 6.14 Trap Port Security

Po spuštění programu **DSNIFF,** jenž svým počtem vygenerovaných MAC adres způsobil přeplnění CAM tabulky a překročil tak možný povolený počet MAC adres na dané rozhraní, SNMP agent vyhodnotil tuto situaci jako porušení nastavených pravidel a zaslal NMS zprávu trap (Obrázek 6.14). Výhodou tohoto SNMP agentem zaslaného upozornění je, že mi umožní rychle a efektivně rozpoznat například nedovolené fyzické připojení cizího počítače pomocí kabelu na rozhraní přepínače umístěného mimo viditelný dosah administrátora či popisované záměrné přetížení systému.

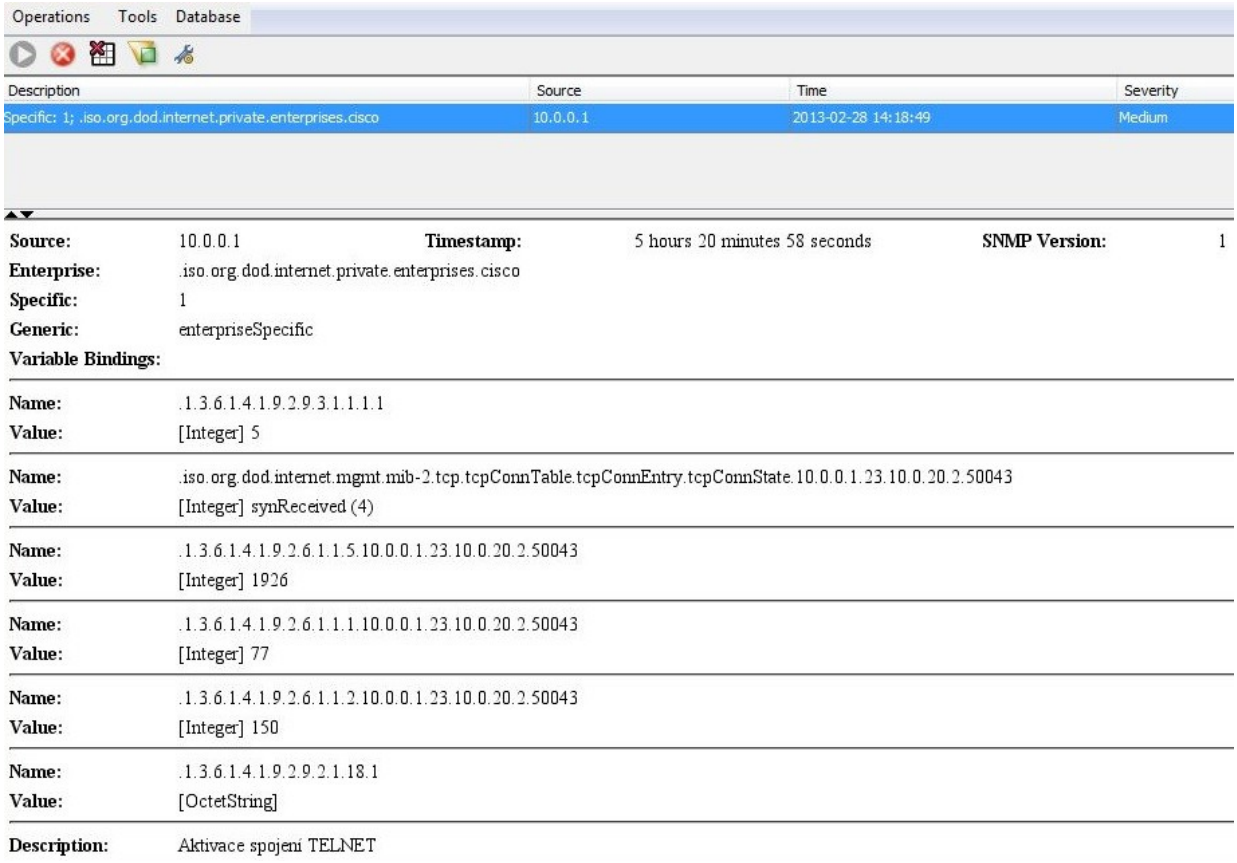
Další, v pořadí třetí simulovaný trap, který v této podkapitole popíši, je trap upozorňující na záměrné vymazání nebo vytvoření sítě VLAN na přepínači *Switch1*. Tato situace může být realizovaná útokem spadajícím pod označení VLAN hopping. Takzvané poskakování po VLAN síti je útok, kdy se útočník snaží dostat k provozu, který využívá pro svůj přenos VLAN sítě. Může tak nastat situace, že útočník záměrně vymaže síť VLAN vytvořenou administrátorem a způsobí tak kolaps, jenž může mít za následek nefunkčnost části sítě. Pro vytvoření tohoto útoku jsem využil dostupný program **Yersinia**, který se zaměřuje na napadání druhé vrstvy síťového protokolu. Po úspěšném spuštění programu **Yersinia** a nastavení procesu vytvoření a vymazání VLAN 150 jsem na NMS obdržel následující trap (Obrázek 6.15).



Obrázek 6.15 Trap vytvoření VLAN150

Díky trapu zaslaném SNMP agentem snadno vyhodnotím situaci a rozpoznám, že došlo k vytvoření a smazání sítě VLAN150. Myslím si, že tato informace je jasnou indicií o tom, že došlo k narušení bezpečnosti tím, že například potencionální útočník vytvořil na velmi krátkou dobu novou síť VLAN, připojil svůj počítač fyzicky do sítě, stáhnul potřebná data a vytvořenou VLAN poté ihned smazal, aby se vyvaroval zaznamenání této činnosti administrátorem.

Poslední, tedy čtvrtý uměle vytvořený trap, který jsem v rámci testovací sítě vytvořil, je narušení bezpečnosti způsobené nedovoleným připojením se pomocí protokolu telnet na použitý prvek sítě s označením *Switch1*, tedy prvek přepínač. Telnet je prostředek, umožňující administrátorovi připojit se k vzdálenému uzlu sítě. Na obrázku 6.16 můžeme vidět trap zprávu zaslanou SNMP agentem *Switch1*.



Obrázek 6.16 Trap telnet spojení

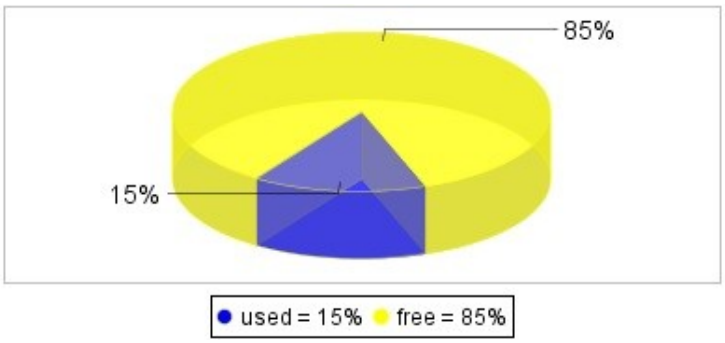
Situací, kde právě použití SNMP trap zpráv administrátor ocení, je velké množství. Za pomocí programu iReasoning MIB Browser je možné nastavit různé případy zasílání trapů a jejich identifikaci či filtraci na základě dané situace nebo konkrétní činnosti, kterou je nutné monitorovat. Administrátor by však měl zvážit, kdy je opravdu nutné nastavovat zasílání trap zpráv dohlížejících na konkrétní činnost generovanou SNMP agentem, a vyvarovat se tak zbytečnému zahlcování sítě nedůležitými informacemi.

Chtěl bych také zmínit, že jsem použitím aplikace SNMP Object Navigator, kterou lze nalézt na internetových stránkách společnosti CISCO, dohledal veškerá číselná OID označení pro konkrétně hledaný objekt MIB databáze a ulehčil si tak práci při orientaci v mnohdy dlouhém OID zápisu. Na závěr této podkapitoly bych chtěl zdůraznit, že všechny popisované situace byly vytvořeny a demonstrovány bez jakéhokoliv nastaveného zabezpečení na použitých prvcích. Svými příklady jsem chtěl popsat vzniklé omezení a dopady na funkčnost sítě s možností snadné identifikace vzniklé příčiny administrátorem.

6.3 Grafy zatížení použitých prvků

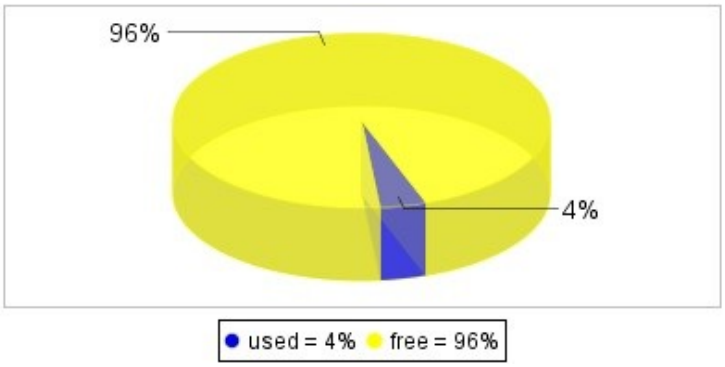
Program iReasoning MIB Browser umožňuje vyhodnocení informací svých SNMP agentů formou grafů pouze pro paměť RAM, CPU a jednotlivá rozhraní monitorovaného prvku. Modře označená plocha grafu představuje využitou část a oblast vybarvená žlutě představuje část nevytíženou, tedy část ať už paměti RAM, nebo CPU, jež je v monitorovaném prvku k dispozici.

Po celou dobu testování praktické části prostřednictvím zvoleného přepínače se hodnota využití paměti RAM u tohoto zařízení neměnila a setrvávala na hodnotě 15 % (Obrázek 6.17).



Obrázek 6.17 Využití RAM přepínače Switch1

Využití CPU přepínače lze vidět na obrázku 6.18. Hodnota 4% využití CPU se neměnila po celou dobu nastavování přepínače či při použití běžných funkcí, jako je *Get* nebo *Set*.

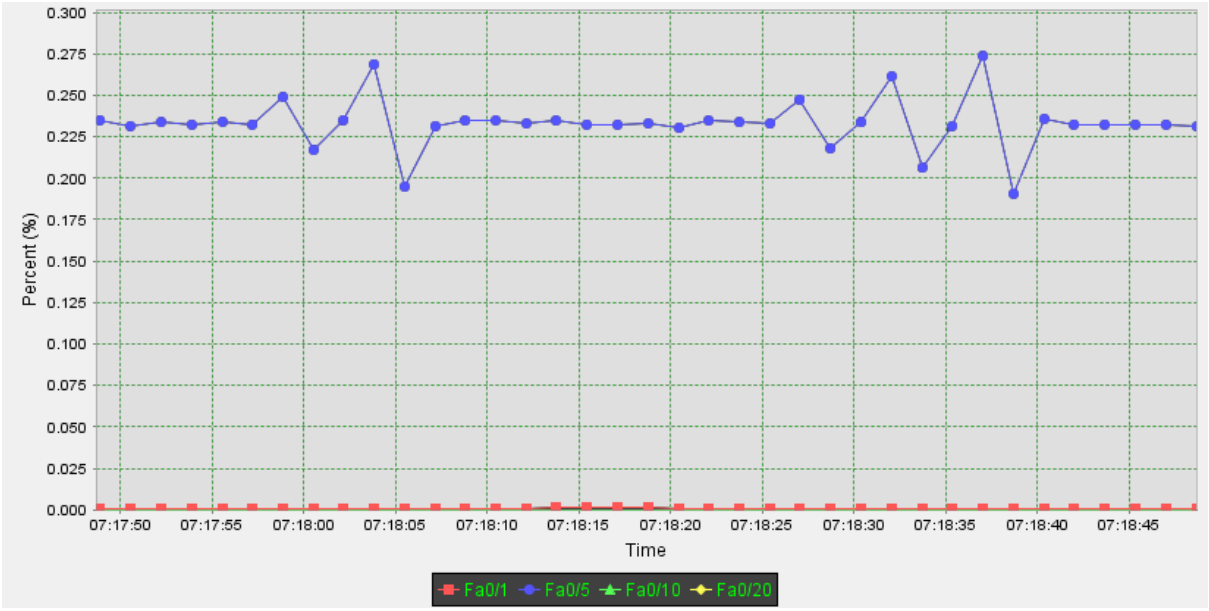


Obrázek 6.18 Využití CPU přepínače Switch1

Zaznamenanou aktivitu portů přepínače *Switch1* lze vidět na obrázku 6.19. V grafu můžeme rozlišit označení jednotlivých křivek, které zvolenou barvou reprezentují daný port. Osa **x** reprezentuje jednotku času (h.min:s), kdy došlo k požadavku monitorování dané aktivity, a osa **y** udává informaci o procentuálním využití konkrétního rozhraní přepínače.

Z grafu lze snadno zjistit, že aktivní byl pouze port Fa0/5, který byl použit pro připojení NMS do sítě. Důvodem neaktivnosti ostatních monitorovaných rozhraní byla nečinnost stanic, které byly na tyto rozhraní připojeny. Můžeme považovat toto vyhodnocení jako jakýsi klidový stav v síti, kdy

dochází pouze k aktivitě NMS s využitím rozhraní přepínače přibližně 0,25 % z důvodu posílání dotazů na své SNMP agenty.



Obrázek 6.19 Aktivita portů nezatíženého prvku Switch1

Pokud se vrátím k situaci, kdy jsem pomocí programu **DSNIFF** zahlcoval rozhraní přepínače *Switch1* uměle vytvořenými MAC adresami, zaznamenal jsem také chování hardwaru testovaného přepínače. Nechal jsem program **DSNIFF** posílat vygenerované MAC adresy na port přepínače Fa0/20 (spojení s PC3) po dobu 5 minut.

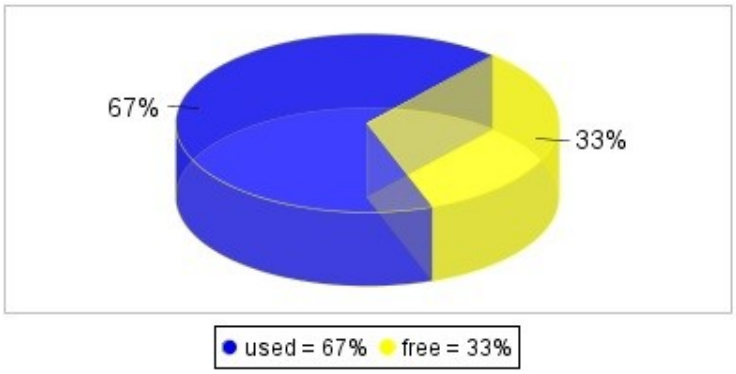
```
Switch1#sh port-security interface fastEthernet 0/20
Port Security              : Enabled
Port Status                : Secure-up
Violation Mode             : Restrict
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 10
Total MAC Addresses        : 10
Configured MAC Addresses   : 0
Sticky MAC Addresses       : 0
Last Source Address:Vlan   : f054.357d.c08b:20
Security Violation Count   : 1011257

Switch1#
```

Obrázek 6.20 Počet vygenerovaných MAC adres pro Switch1

Na obrázku 6.20 je pomocí Cisco IOS zobrazen výpis port-security pro rozhraní Fa0/20 přepínače *Switch1*, jelikož tuto funkci použitý program iReasoning MIB Browser nepodporuje. Je zřejmé, že za dobu, kdy byl program DSNIFF spuštěn, došlo 1 011 257krát k porušení bezpečnosti

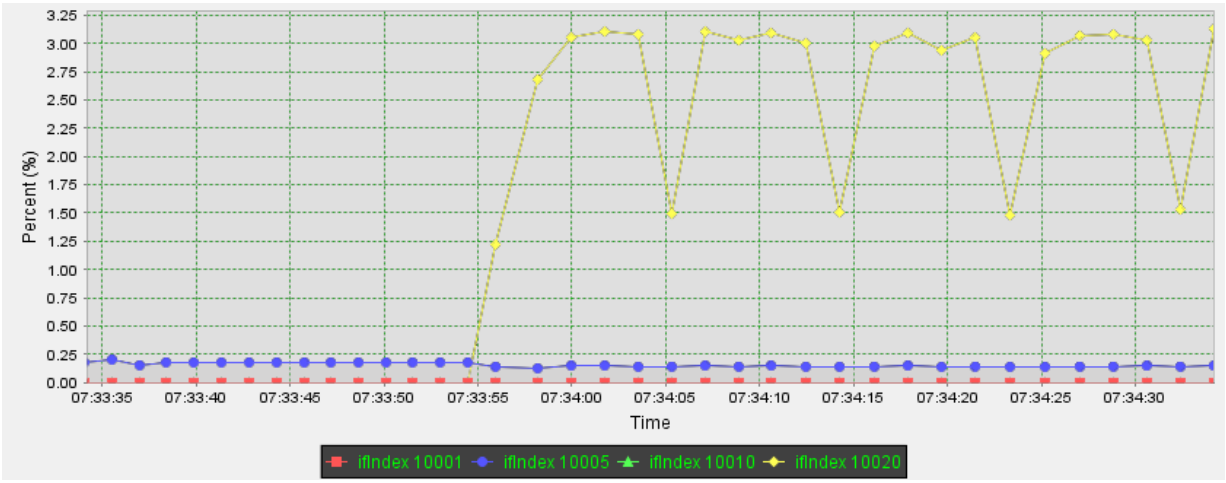
(nepropuštění přenosu dat rozhraním z neznámé MAC adresy). V tomto časovém intervalu program iReasoning MIB Browser vytvořil graf zatížení paměti RAM a CPU přepínače.

Hodnoty využití RAM paměti a CPU monitorovaného prvku se začaly měnit, když jsem spustil na PC2 program **DSNIFF.** Dle očekávání výkon CPU přepínače postupně vzrůstal. Po pěti minutách nepřetržitého generování uměle vytvořených MAC adres zasílaných na přepínač *Switch1* byla zaznamenána hodnota využití CPU již 67 % (Obrázek 6.21).



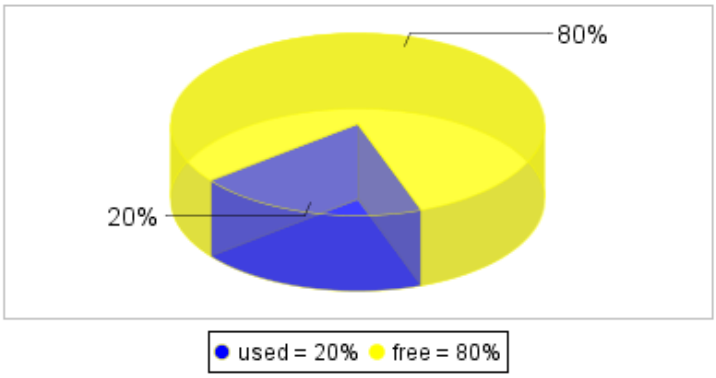
Obrázek 6.21 Nejvyšší zaznamenané využití CPU přepínače Switch1

Prudký nárůst činnosti rozhraní Fa0/20 je zachycen na obrázku 6.22. Z výsledků je patrné, že i přes simulovaný DoS útok aktivita rozhraní *Switch1* nepřesáhla hodnotu 3,25 % svého vytížení.



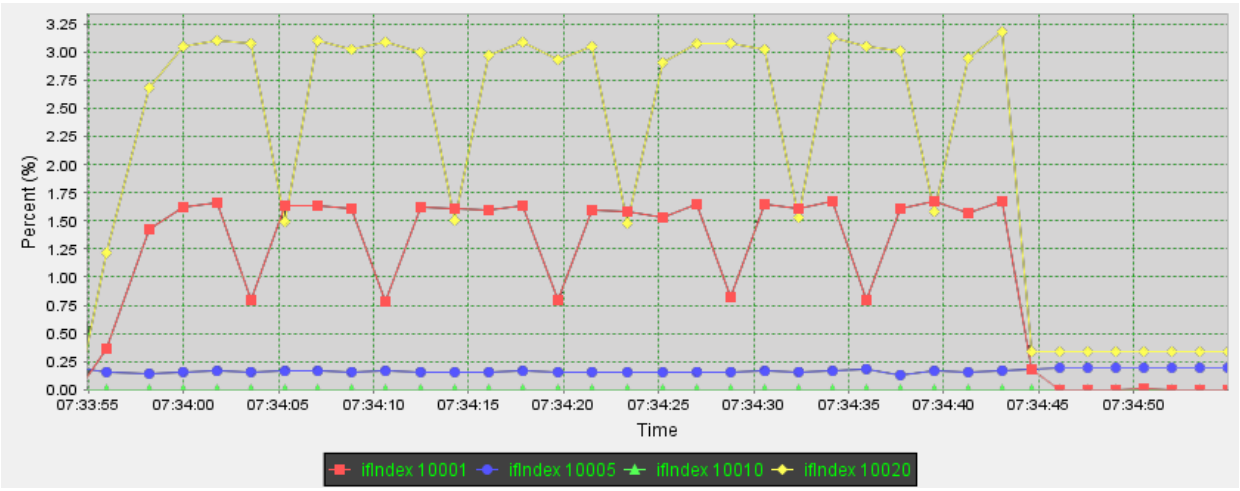
Obrázek 6.22 Nárůst činnosti rozhraní přepínače Switch1 při MAC address flooding

V praktické části, kterou zde popisuji, jsem k zaznamenání aktivity na jednotlivých rozhraních síťových prvků vytvořil FTP přenos. V rámci použité topologie byl zvolen PC3 jako FTP server a PC1 jako FTP klient. Pomocí PC1 jsem tedy stahoval soubor, který byl uložen na PC3. Během přenosu byl zaznamenán nárůst využití paměti RAM přepínače na úroveň 20 % (Obrázek 6.23). Hodnota využití CPU přepínače se neměnila a setrvávala tak na úrovni 4 %.



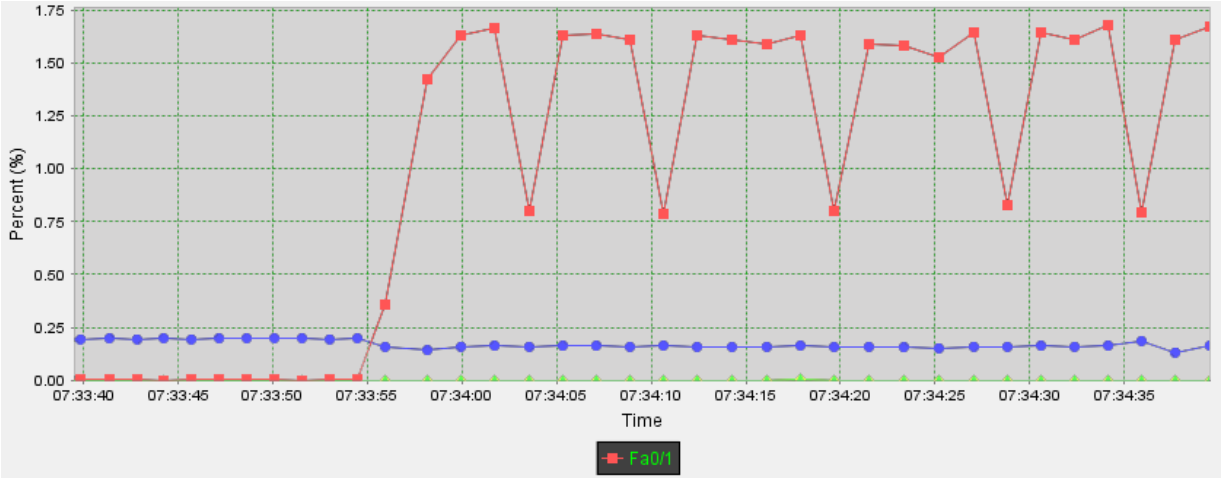
Obrázek 6.23 Využití paměti RAM přepínače Switch1 při FTP přenosu

Díky tomu, že program umožňuje v pravidelných intervalech získávat informace o kterémkoli zařízení podporujícím SNMP protokol, je tak možné vytvořit graf zobrazující požadovanou aktivitu. FTP přenos probíhal po dobu 50 sekund s tím, že došlo k odeslání části 700MB souboru. Důvod, proč jsem celý soubor nepřenesl, je ten, že jsem chtěl zachytit takzvané aktuální parametry sítě. I když prohlížením těchto údajů mnoho informací o dlouhodobějším chodu sítě nezískáme, vidím výhodu zvoleného postupu v tom, že díky aktuálním parametrům sítě může administrátor s velkou přesností a v daný okamžik analyzovat její stav.



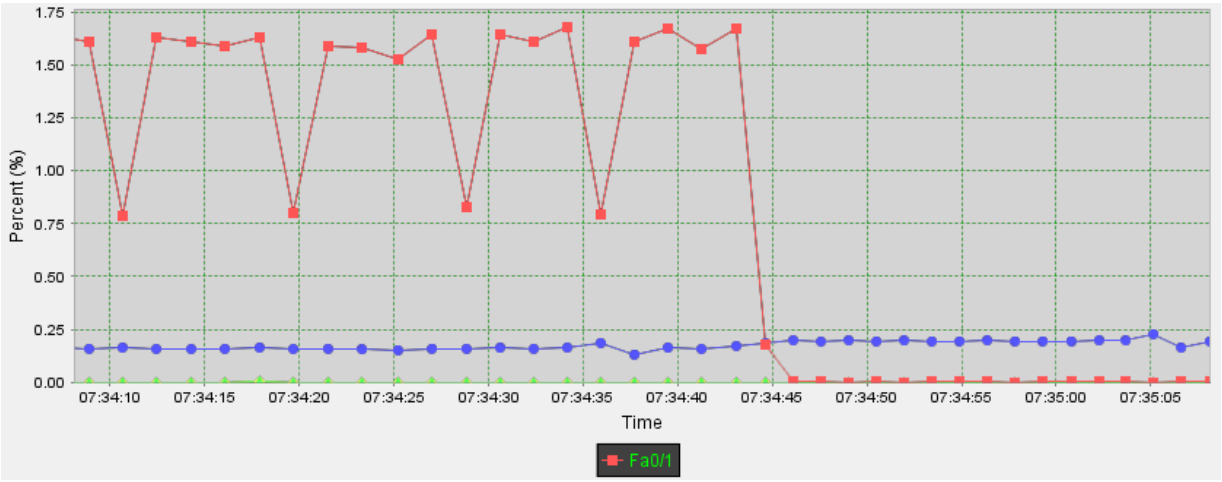
Obrázek 6.24 FTP přenos souboru Switch1

Základním dotazovacím časovým intervalem je 5 sekund, tedy každých 5 sekund vyšle NMS svůj požadavek za účelem získání potřebných údajů od svého SNMP agenta. Na obrázku 6.24 je vidět průběh přenosu souboru. Během tohoto přenosu byla zaznamenána aktivita přepínače *Switch1* na portech Fa0/1, použitém jako spojení TRUNK mezi *Switch1* a *Router2,* s maximálním 1,75% využitím rozhraní a Fa0/20 (spojení s PC3) s maximální hodnotou 3,25 % svého vytížení. V čase 07.34:50 byl záměrně FTP přenos přerušen. Pomocí grafu lze popsaný stav zaznamenat okamžitým snížením procentuálního využití monitorovaných portů téměř na hodnotu 0,25 %.



Obrázek 6.25 FTP přenos souboru Router1

Náhlý nárůst aktivity rozhraní Fa0/1 lze vidět i na grafu průběhu přepínače *Router1* (Obrázek 6.25). Na toto rozhraní je připojena stanice *PC1* sloužící jako FTP klient.



Obrázek 6.26 FTP přenos souboru Router2

Doba ukončení FTP přenosu se dle očekávání projevila i na monitorované aktivitě rozhraní Fa0/1 u směrovače *Router2*. Popisované rozhraní (Obrázek 6.26) slouží jako TRUNK spojení mezi *Router2* a *Switch1*.

7 Závěr

V této bakalářské práci se čtenář mohl seznámit s problematikou SNMP protokolu a způsobem, jak se co nejvhodněji dostat k informacím uloženým v databázi MIB testovaných síťových prvků společnosti CISCO. Práce rovněž seznamuje čtenáře s problematikou shromažďování a vyhodnocování informací o provozu ve vytvořené experimentální síti v laboratoři N312.

V první třetině praktické části této bakalářské práce byly popsány postupy práce se základními funkcemi protokolu SNMP. Pomocí základního dotazování se SNMP agentů má administrátor možnost udělat si celkovou představu o nastavení síťových prvků, které hrají klíčovou roli pro bezproblémovou funkčnost a dohled nad celou sítí. Je zde zmíněn například popis rozhraní spojující klíčové prvky sítě, kontrola správnosti přiřazení segmentu počítačové sítě do určené sítě VLAN z důvodu bezpečnosti či mnohdy klíčový údaj týkající se funkčnosti daného rozhraní. Účelná může být také změna kontaktu administrátora, který je v případě poruchy prvku zodpovědný za jeho správu. SNMP protokol umožňuje tuto změnu provést pomocí funkce Set bez nutnosti vstoupení do konfiguračního režimu nastavovaného prvku.

Druhá třetina praktické části je zaměřena na simulaci reálného provozu v různých situacích, které mohou nastat, a na konfiguraci použitého programu určeného pro správu sítě tak, aby pomocí trap zpráv co nejlépe reagoval na vzniklé situace. Jsou zde uvedeny simulace výpadku spoje, upozornění umožňující informovat administrátora o události vzniklé zahlcením portů směrovače prostřednictvím DoS útoku nebo narušení bezpečnosti vytvořením VLAN sítě za účelem stažení citlivých dat bez vědomí administrátora.

Třetí třetina praktické části umožňuje čtenáři nahlédnout do krátkodobých charakteristik a záznamů měření jednotlivých parametrů síťového provozu formou grafů. Jsou zde uvedeny naměřené hodnoty zatížení paměti RAM a procesoru testovaných síťových prvků jak při klidovém stavu v síti bez jakéhokoliv ovlivnění, tak i při vytvořeném DoS útoku nebo grafy zatížení monitorovaných rozhraní prvků při vytvořeném FTP přenosu souboru.

Záměrem této bakalářské práce bylo umožnit čtenáři proniknout do problematiky dohledu a správy sítí pomocí protokolu SNMP, ale hlavně popsat výhody, které s sebou nese nasazení interakce SNMP protokolu a programu určeného pro monitoring sítě. Efektivní správa datových sítí se dnes neobejde bez použití monitorovacího programu, který může administrátorovi pomoci optimalizovat chod sítě a předejít tak poruchám nebo je co nejrychleji lokalizovat a řešit. Jen tak může administrátor zaručit požadovanou kvalitu, spolehlivost a dostupnost spravované datové sítě.

Použitá literatura

1. **Pužmanová, Rita.** *Moderní komunikační sítě od A do Z.* Brno : Computer Press, a.s., 2006.

2. Wikipedia.org. *SNMP.* [Online] 27. Říjen 2012. [Citace: 1. Listopad 2012.] http://en.wikipedia.org/wiki/Snmp.

3. **Klaška, Luboš.** Vznik a principy SNMP. *Svět sítí.* [Online] Svět sítí & Infinity a.s., 11. Červen 2000. [Citace: 2. Listopad 2012.] http://www.svetsiti.cz/clanek.asp?cid=Vznik-a-principy-SNMP-1162000.

4. Wikipedia. *Simple Gateway Monitoring Protocol.* [Online] 20. 4 2010. http://en.wikipedia.org/wiki/Simple_Gateway_Monitoring_Protocol.

5. Wikipedia. *Common Management Information Protocol.* [Online] 30. Červenec 2012. http://en.wikipedia.org/wiki/Common_Management_Information_Protocol.

6. docwiki.cisco.com. *Simple Network Management Protocol.* [Online] 16. Říjen 2012, http://docwiki.cisco.com/wiki/Simple_Network_Management_Protocol#Figure:_An_SNMP-Managed_Network_Consists_of_Managed_Devices.2C_Agents.2C_and_NMSs.

7. **Douglas, Bruey.** SNMP: Simple? Network Management Protocol. *Rane.* [Online] Rane Corporation, Prosinec 2005. http://www.rane.com/note161.html.

8. **Tunstall, Craig a Cole, Gwyn.** *Developing WMI Solutions: A Guide to Windows Management Instrumentation.* Boston : Addison-Wesley Professional, 2012. 0-201-61613-0.

9. **Klaška, Luboš.** Svět sítí. *Formát SNMP zpráv.* [Online] 14. Červen 2000. http://www.svetsiti.cz/clanek.asp?cid=Format-SNMP-zprav-1462000.

10. **Zeltserman, Dave.** *Practical Guide to Snmpv3 and Network Management.* New Jersey : Prentice Hall, 1999. 0-13-021453-1.

11. **Stallings, William.** Cisco. *Security Comes to SNMP: The New SNMPv3 Proposed Internet Standards.* [Online] Cisco. http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_1-3/snmpv3.html.

12. **Schmidt, Kevin a Mauro, Douglas.** *Essential SNMP, 2nd Edition.* Sebastopol : O'Reilly Media, 2005. 978-0-596-00840-6.

13. **Klaška, Luboš.** Svet siti. *Další vývoj protokolu SNMP.* [Online] 15. Červen 2000. http://www.svetsiti.cz/clanek.asp?cid=Dalsi-vyvoj-protokolu-SNMP-1562000.

14. Wikipedia.org. *Management information base.* [Online] Wikipedia, 31. Srpen 2012. http://en.wikipedia.org/wiki/Management_information_base.

15. **McCloghrie, K., a další.** RFC 2578. *Structure of Management Information Version 2.* [Online] Duben 1999. http://www.ietf.org/rfc/rfc2578.txt.

16. **Velte, Toby J. a Velte, Antony T.** *Síťové technologie Cisco: Velký průvodce.* Brno : Computer Press, 2003. 80-7226-857-0.

17. **Cisco.** Cisco. *Cisco Management Information Base (MIB) User Quick Reference.* [Online] Cisco. http://www.cisco.com/en/US/docs/ios/11_2/mib/quick/reference/mover.html.

Seznam obrázků

Seznam tabulek

Seznam příloh

Součástí BP je CD.

Adresářová struktura přiloženého CD:

\Bakalarska_prace_CZY017\

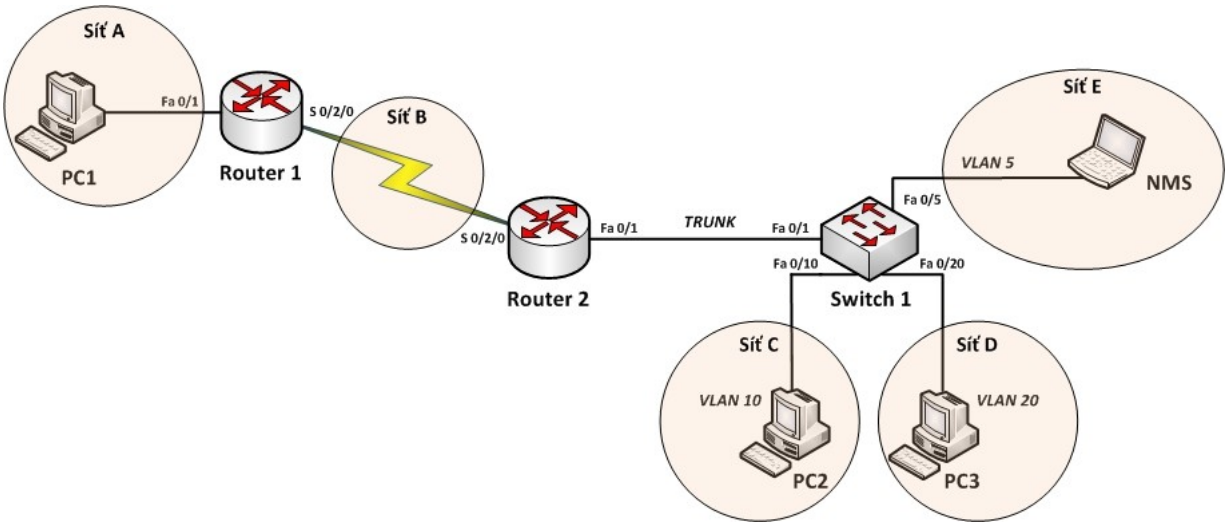
\Bakalarska_prace

- Přiložená bakalářská práce ve formátu PDF.

\Grafy

- Přiložené grafy v grafickém formátu PNG.

Příloha.A: Nastavení jednotlivých síťových prvků pomocí CISCO IOS příkazů



PC1

ifconfig eth1 10.0.40.2 netmask 255.255.255.0

route add default gw 10.0.40.1

PC2

ifconfig eth1 10.0.10.2 netmask 255.255.255.0

route add default gw 10.0.10.10

PC3

ifconfig eth1 10.0.20.2 netmask 255.255.255.0

route add default gw 10.0.20.10

NMS

ifconfig eth1 10.0.0.100 netmask 255.255.255.0

route add default gw 10.0.0.10

Switch1

Switch>enable

Switch#conf terminal

Switch(config)#hostname Switch1

Switch1(config)#no ip domain lookup

Switch1(config)#banner motd :Toto je Switch1:

Switch1(config)#line console 0

Switch1(config-line)#password michal

```
Switch1(config-line)#login
Switch1(config-line)#logging synchronous
Switch1(config-line)#exit

Switch1(config)#line vty 0 15
Switch1(config-line)#password michal
Switch1(config-line)#login
Switch1(config-line)#exit

Switch1(config)#vlan 5
Switch1(config-vlan)#name VLAN5
Switch1(config-vlan)#exit
Switch1(config)#interface vlan 5
Switch1(config-if)#ip address 10.0.0.1 255.255.255.0
Switch1(config-if)#exit

Switch1(config)#vlan 10
Switch1(config-vlan)#name VLAN10
Switch1(config-vlan)#exit
Switch1(config)#interface vlan 10
Switch1(config-if)#ip address 10.0.10.1 255.255.255.0
Switch1(config-if)#exit

Switch1(config)#vlan 20
Switch1(config-vlan)#name VLAN20
Switch1(config-vlan)#exit
Switch1(config)#interface vlan 20
Switch1(config-if)#ip address 10.0.20.1 255.255.255.0
Switch1(config-if)#exit

Switch1(config)#interface fastEthernet 0/5
Switch1(config-if)#switchport mode access
Switch1(config-if)#switchport access vlan 5
Switch1(config-if)#description Spojeni Switch1-NMS
Switch1(config-if)#no shutdown
Switch1(config-if)#exit

Switch1(config)#interface fastEthernet 0/10
Switch1(config-if)#switchport mode access
Switch1(config-if)#switchport access vlan 10
Switch1(config-if)#description Spojeni Switch1-PC2
```

Switch1(config-if)#no shutdown
Switch1(config-if)#exit

Switch1(config)#interface fastEthernet 0/20
Switch1(config-if)#switchport mode access
Switch1(config-if)#switchport access vlan 20
Switch1(config-if)#description Spojeni Switch1-PC3
Switch1(config-if)#switchport port-security
Switch1(config-if)#switchport port-security maximum 10
Switch1(config-if)#switchport port-security violation restrict
Switch1(config-if)#no shutdown
Switch1(config-if)#exit

Switch1(config)#interface fastEthernet 0/1
Switch1(config-if)#description Spojeni TRUNK Switch1-Router2
Switch1(config-if)#switchport mode trunk
Switch1(config-if)#exit

Switch1(config)#snmp-server community vsb RW
Switch1(config)#snmp-server host 10.0.0.100 version 1 vsb
Switch1(config)#snmp-server location VSB-TUO
Switch1(config)#snmp-server contact Administrator
Switch1(config)#snmp-server chassis-id Cisco Catalyst 2960 Series
Switch1(config)#snmp-server enable traps snmp [authentication] [linkdown] [coldstart] [warmstart]
Switch1(config)#snmp-server enable traps tty
Switch1(config)#snmp-server enable traps vlancreate
Switch1(config)#snmp-server enable traps vlandelete
Switch1(config)#snmp-server enable traps port-security
Switch1(config)#snmp-server enable traps config
Switch1(config)#exit
Switch1#copy running-config startup-config

Router2
Router>enable
Router#conf terminal
Router(config)#hostname Router2
Router2(config)#no ip domain lookup
Router2(config)#banner motd :Toto je Router2:

Router2(config)#line console 0
Router2(config-line)#password michal

```
Router2(config-line)#login
Router2(config-line)#logging synchronous
Router2(config-line)#exit

Router2(config)#line vty 0 4
Router2(config-line)#password michal
Router2(config-line)#login
Router2(config-line)#exit

Router2(config)#interface serial 0/2/0
Router2(config-if)#description Spojeni Router2-Router1
Router2(config-if)#ip address 10.0.30.1 255.255.255.252
Router2(config-if)#clock rate 1000000
Router2(config-if)#no shutdown
Router2(config-if)#exit

Router2(config)#interface fastEthernet 0/1
Router2(config-if)#description Spojeni Router2-Switch1
Router2(config-if)#no shutdown
Router2(config-if)#exit
Router2(config)#interface fastEthernet 0/1.5
Router2(config-subif)#encapsulation dot1Q 5
Router2(config-subif)#ip address 10.0.0.10 255.255.255.0
Router2(config-subif)#exit

Router2(config)#interface fastEthernet 0/1.10
Router2(config-subif)#encapsulation dot1Q 10
Router2(config-subif)#ip address 10.0.10.10 255.255.255.0
Router2(config-subif)#exit

Router2(config)#interface fastEthernet 0/1.20
Router2(config-subif)#encapsulation dot1Q 20
Router2(config-subif)#ip address 10.0.20.10 255.255.255.0
Router2(config-subif)#exit

Router2(config)#ip route 10.0.40.0 255.255.255.0 10.0.30.2

Router2(config)#snmp-server community vsb RW
Router2(config)#snmp-server host 10.0.0.100 version 1vsb
Router2(config)#snmp-server location VSB-TUO
Router2(config)#snmp-server contact Administrator
```

Router2(config)#snmp-server chassis-id Cisco 2800 Series

Router2(config)#snmp-server enable traps snmp [authentication] [linkdown] [coldstart] [warmstart]

Router2(config)#snmp-server enable traps config

Router2(config)#snmp-server enable traps tty

Router2(config)#exit

Router2#copy running-config startup-config

Router1

Router>

Router>enable

Router#conf terminal

Router(config)#hostname Router1

Router1(config)#no ip domain lookup

Router1(config)#banner motd :Toto je Router1:

Router1(config)#line console 0

Router1(config-line)#password michal

Router1(config-line)#login

Router1(config-line)#logging synchronous

Router1(config-line)#exit

Router1(config)#line vty 0 4

Router1(config-line)#password michal

Router1(config-line)#login

Router1(config-line)#exit

Router1(config)#interface serial 0/2/0

Router1(config-if)#description Spojeni Router1-Router2

Router1(config-if)#ip address 10.0.30.2 255.255.255.252

Router1(config-if)#no shutdown

Router1(config-if)#exit

Router1(config)#interface fastEthernet 0/1

Router1(config-if)#description Spojeni Router1-PC1

Router1(config-if)#ip address 10.0.40.1 255.255.255.0

Router1(config-if)#no shutdown

Router1(config-if)#exit

Router1(config)#ip route 10.0.0.0 255.255.255.0 10.0.30.1

Router1(config)#ip route 10.0.10.0 255.255.255.0 10.0.30.1

Router1(config)#ip route 10.0.20.0 255.255.255.0 10.0.30.1

```
Router1(config)#snmp-server community vsb RW
Router1(config)#snmp-server host 10.0.0.100 version 1vsb
Router1(config)#snmp-server location VSB-TUO
Router1(config)#snmp-server contact Administrator
Router1(config)#snmp-server chassis-id Cisco 2800 Series
Router1(config)#snmp-server enable traps snmp [authentication] [linkdown] [coldstart] [warmstart]
Router2(config)#snmp-server enable traps config
Router2(config)#snmp-server enable traps tty
Router1(config)#exit
Router1#copy running-config startup-config
```

Příloha.B: Použité rozsahy IP adres pro sítě a rozhraní jednotlivých prvků

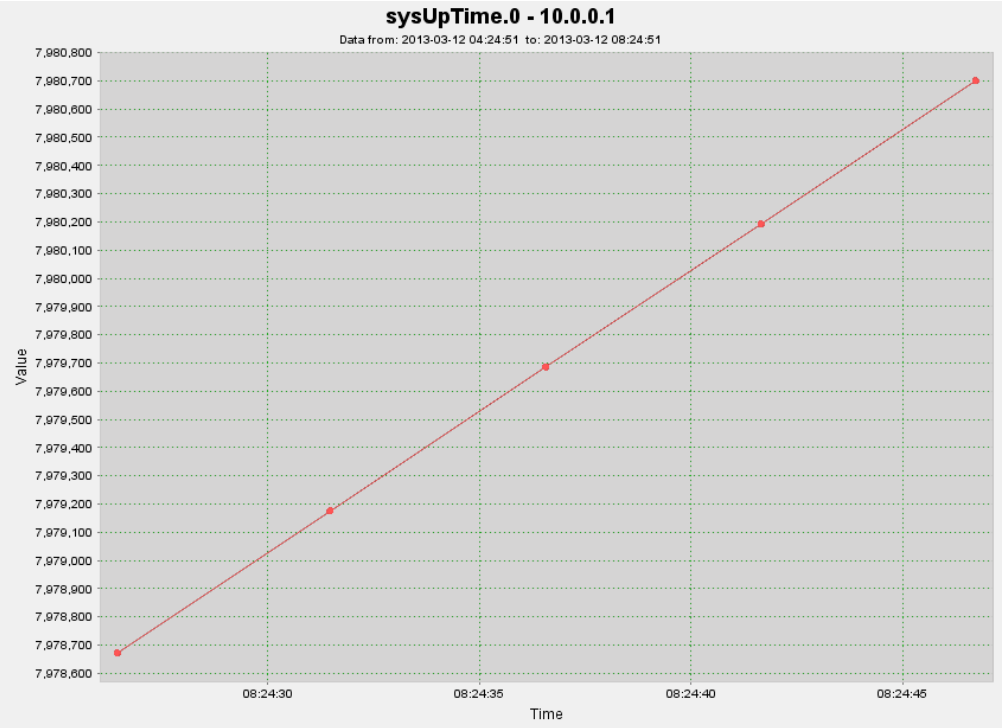
Rozsahy IP adres pro jednotlivé sítě:

	Adresa sítě
Síť A	*10.0.40.0/24*
Síť B	*10.0.30.0/30*
Síť C	*10.0.10.0/24*
Síť D	*10.0.20.0/24*
Síť E	*10.0.0.0/24*
R2 - SW1	*trunk*

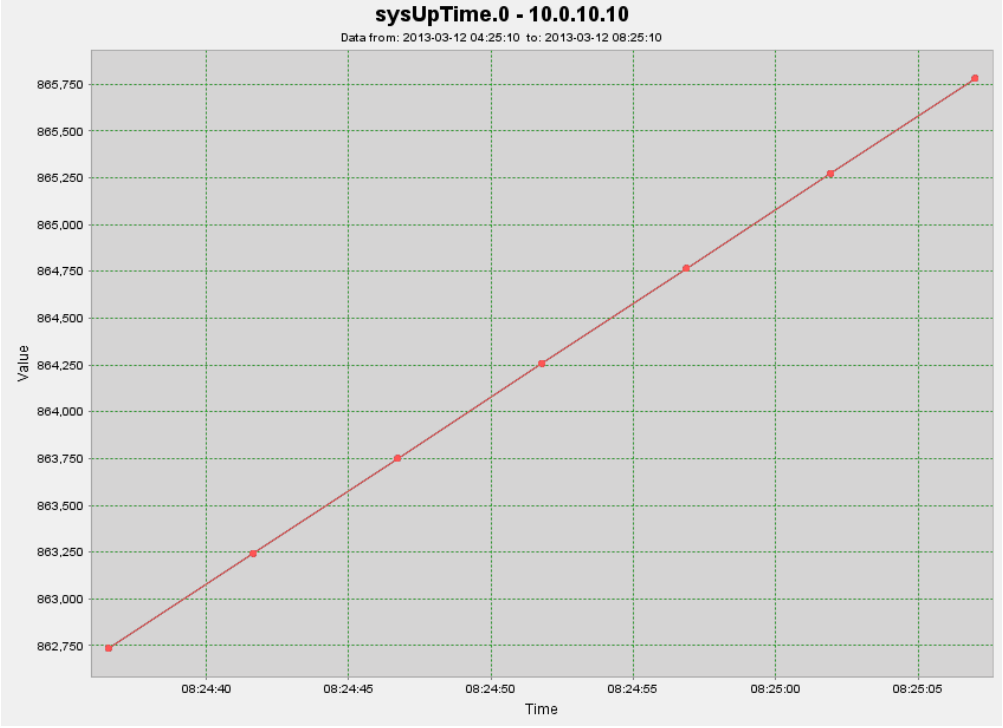
Přidělené IP adresy na rozhraní použitých prvků:

	Rozhraní	**IP adresa**
Router 1	*Fa 0/1*	*10.0.40.1*
Router 1	*S 0/2/0*	*10.0.30.2*
Router 2	*Fa 0/1*	*Trunk*
Router 2	*S 0/2/0*	*10.0.30.1*
Switch 1	*Fa 0/1*	*Trunk*
Switch 1	*Fa 0/10*	*10.0.10.1*
Switch 1	*Fa 0/20*	*10.0.20.1*
Switch 1	*Fa 0/5*	*10.0.0.1*
PC1	*Eth1*	*10.0.40.2*
PC2	*Eth1*	*10.0.10.2*
PC3	*Eth1*	*10.0.20.2*
NMS	*FastEthernet*	*10.0.0.100*

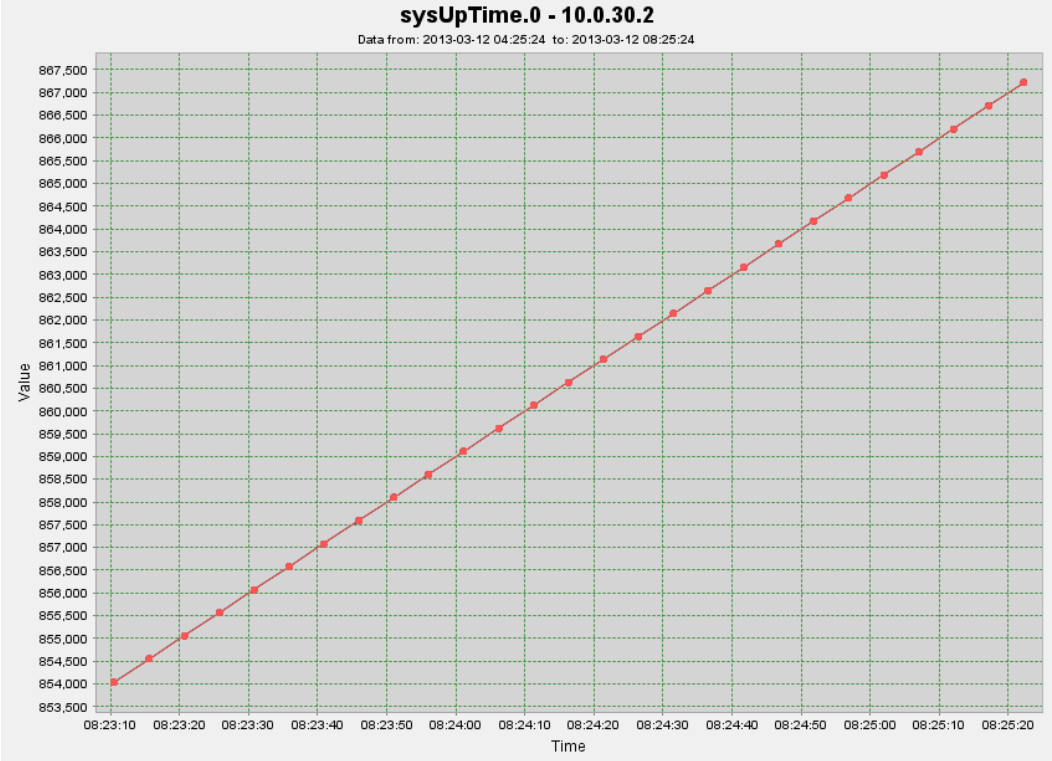
Příloha.C: Zaznamenané grafy monitorující konkrétní událost na popisovaném prvku



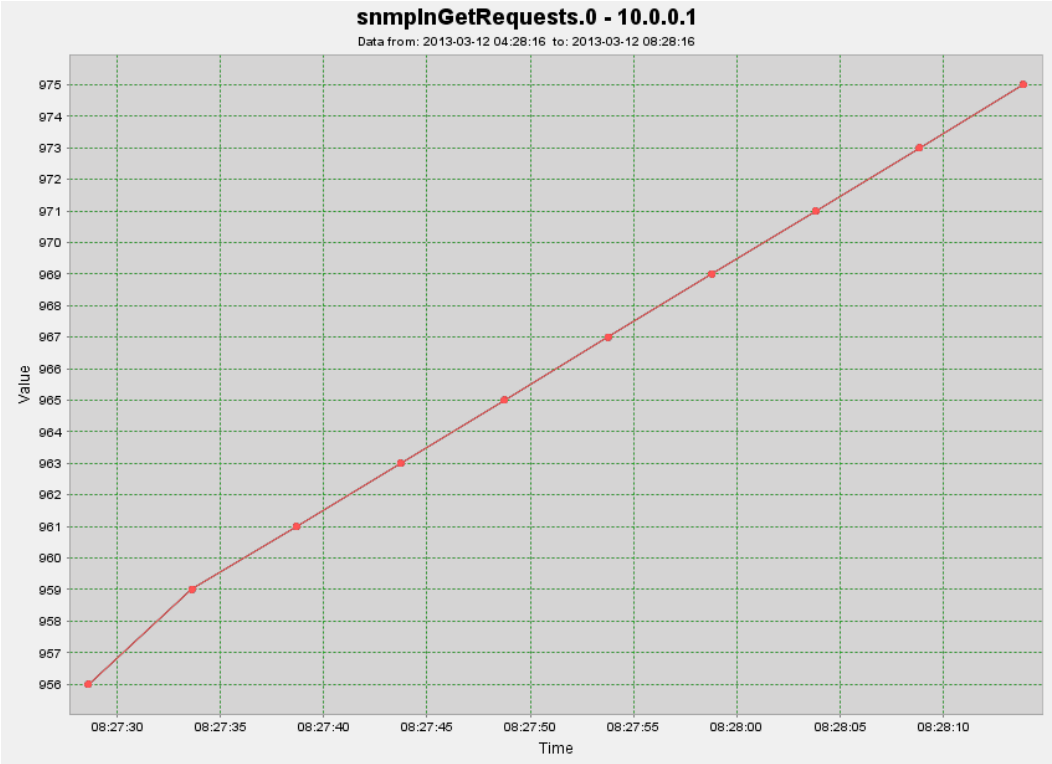
Graf č. 1 Zaznamenaná doba provozu přepínače Switch1 během testování je 22.02:05



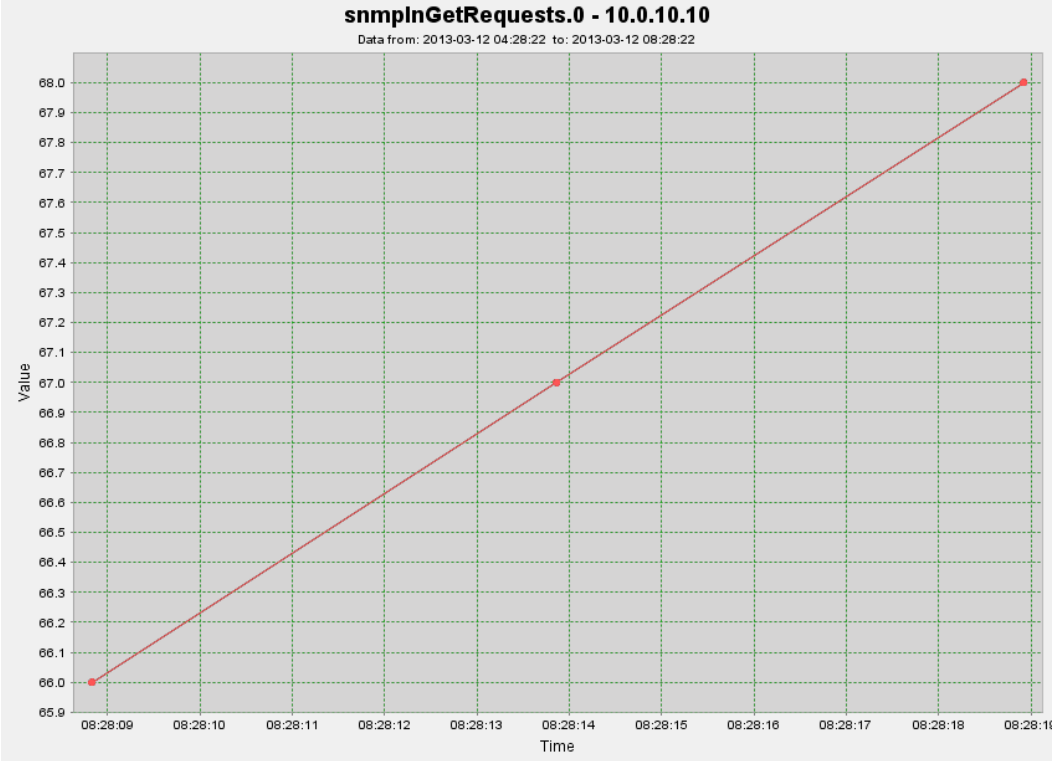
Graf č. 2 Zaznamenaná doba provozu směrovače Router2 během testování je 2.06:00



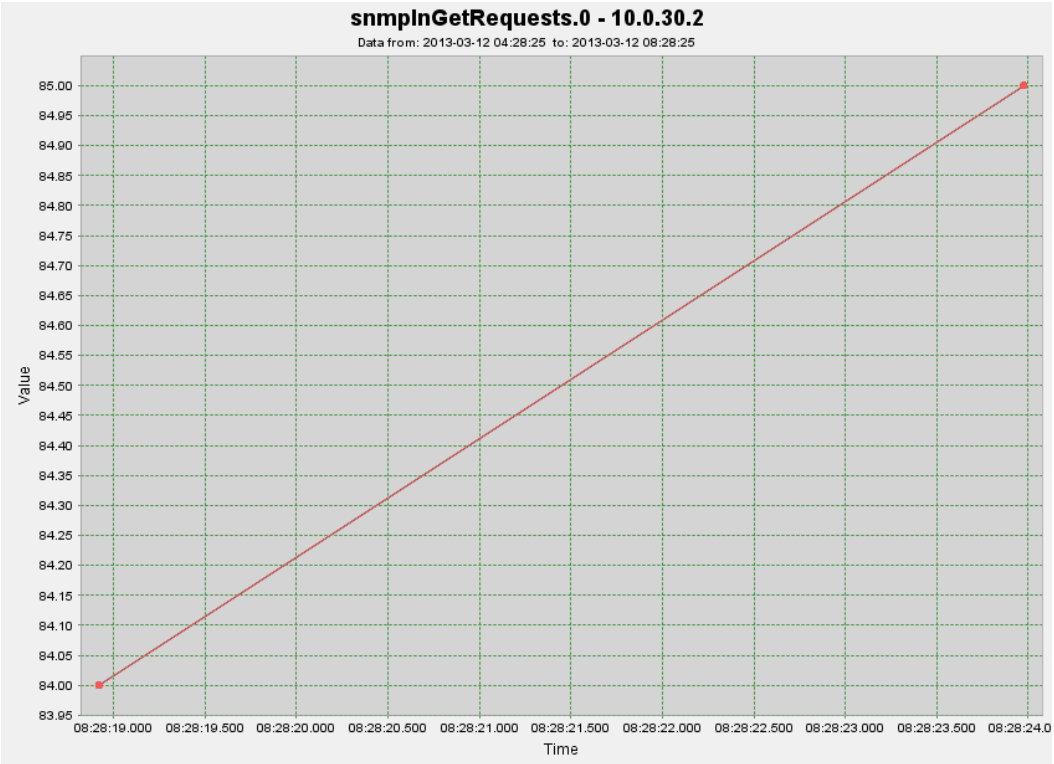
Graf č. 3 Zaznamenaná doba provozu směrovače Router1 během testování je 2.06:01



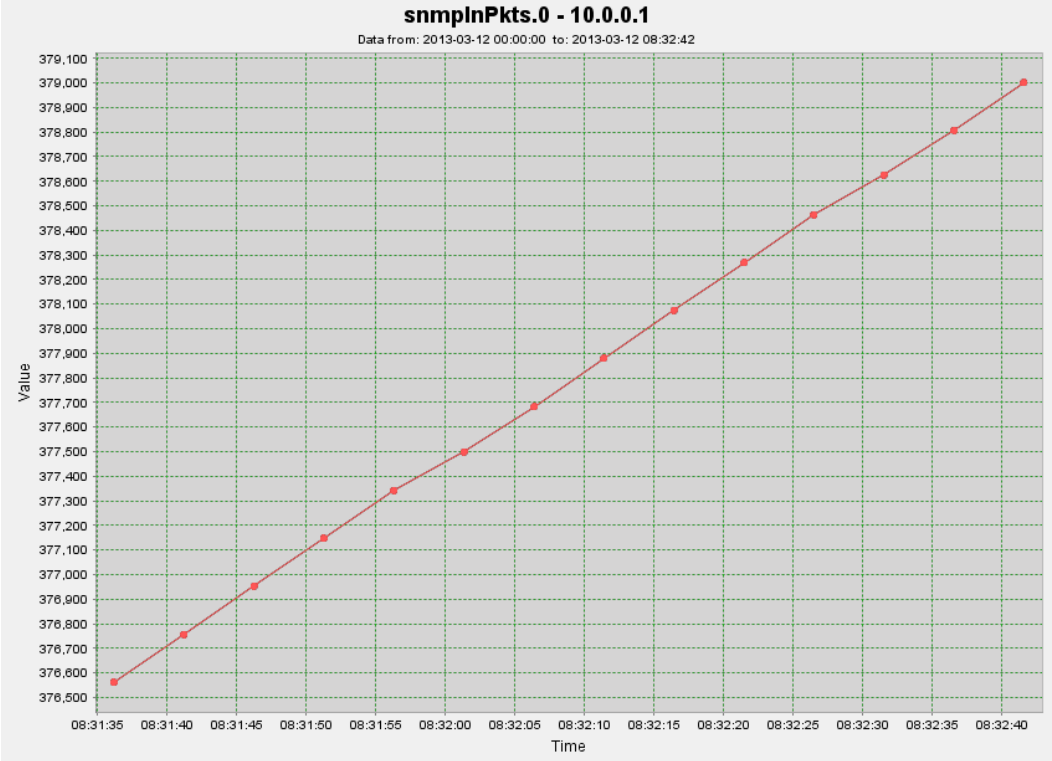
Graf č. 4 Zaznamenaný počet 975 příchozích GetRequest zpráv na přepínač Switch1 během testování



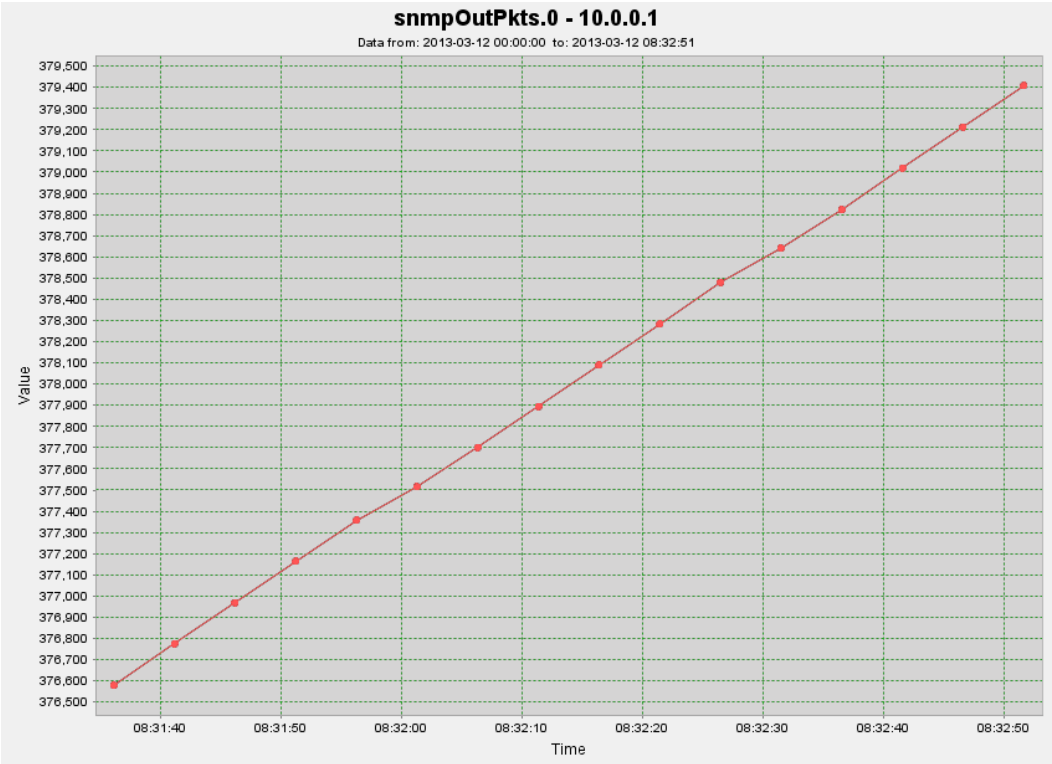
Graf č. 5 Zaznamenaný počet 68 příchozích GetRequest zpráv na směrovač Router2 během testování



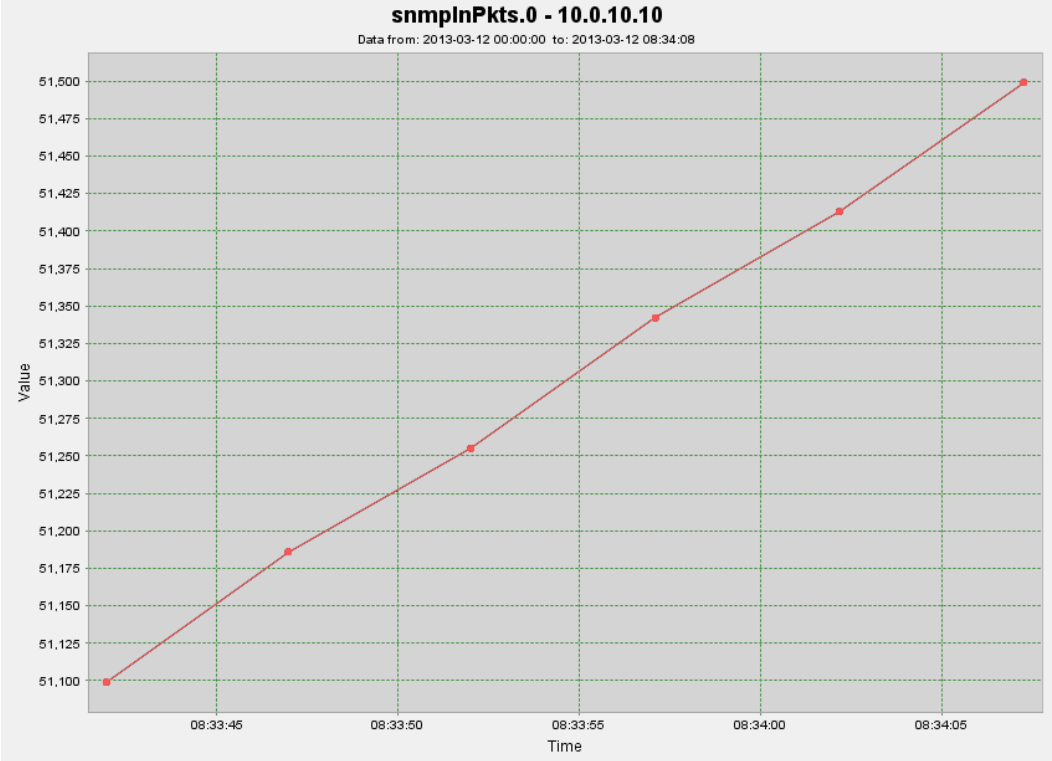
Graf č. 6 *Zaznamenaný počet 85 příchozích GetRequest zpráv na směrovač Router1 během testování*



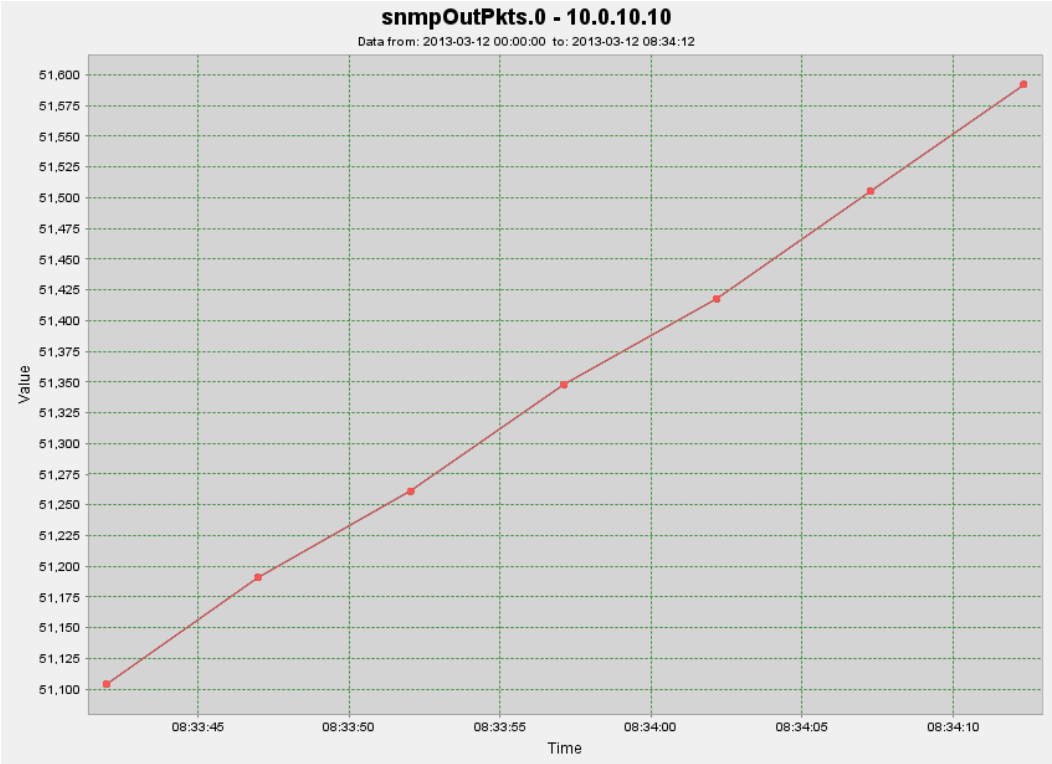
Graf č. 7 Zaznamenaný počet 379 000 příchozích SNMP zpráv přepínače Switch1 během testování



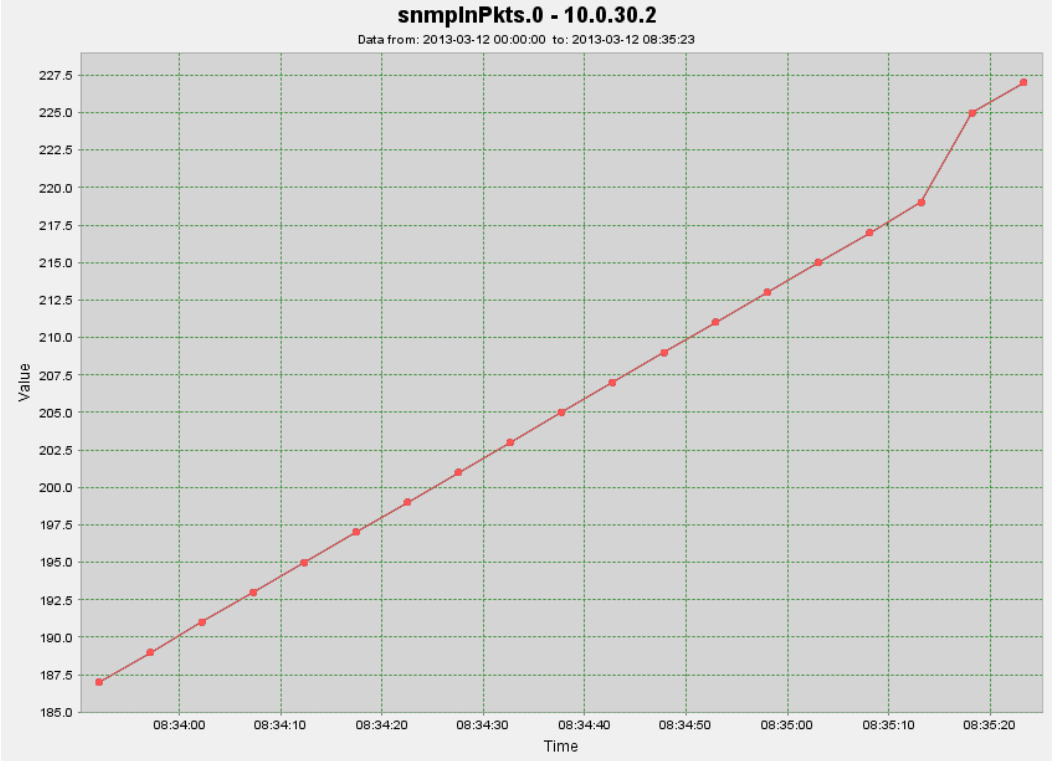
Graf č. 8 Zaznamenaný počet 379 400 odchozích SNMP zpráv přepínače Switch1 během testování



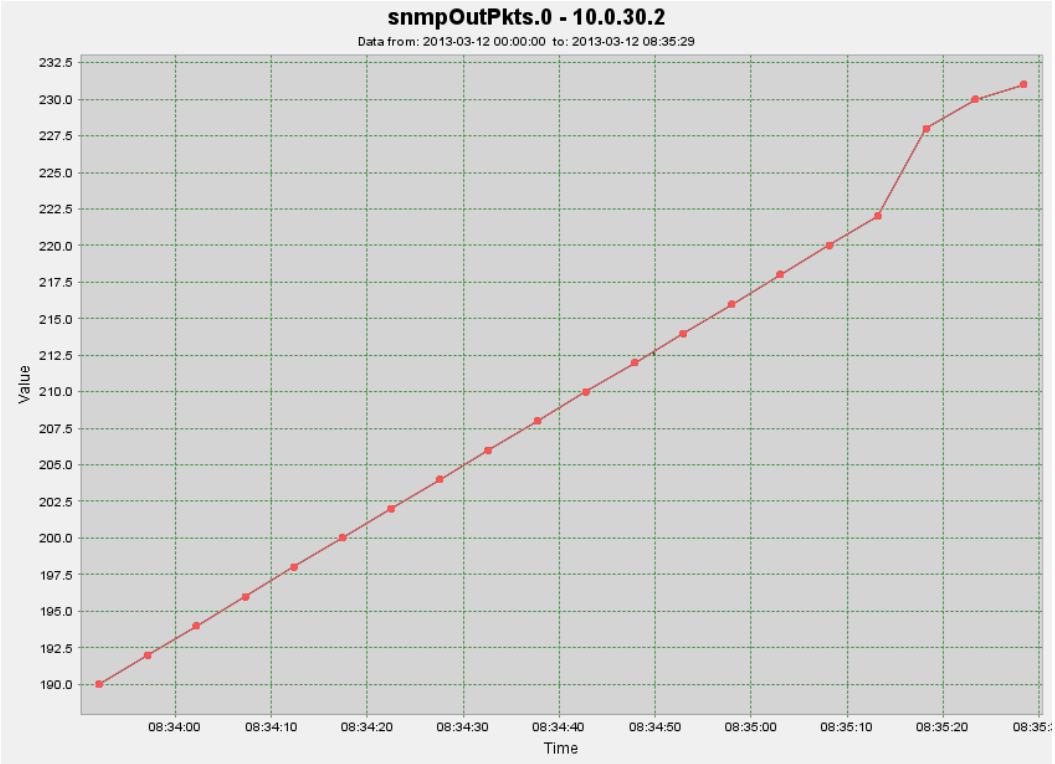
Graf č. 9 Zaznamenaný počet 51 500 příchozích SNMP zpráv směrovače Router2 během testování



Graf č. 10 Zaznamenaný počet 51 600 odchozích SNMP zpráv směrovače Router2 během testování

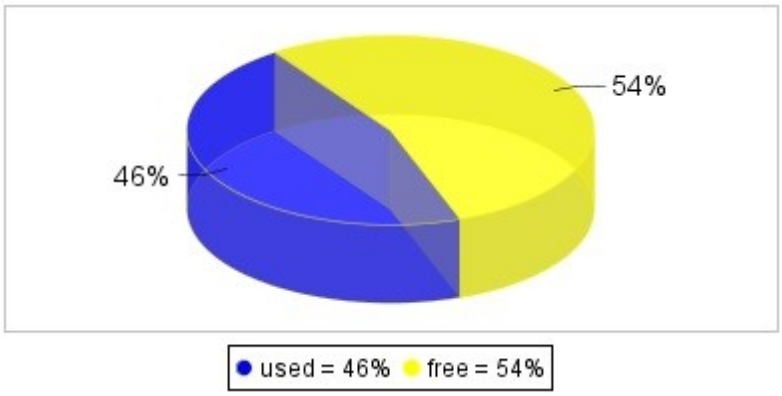


Graf č. 11 Zaznamenaný počet 227 příchozích SNMP zpráv směrovače Router1 během testování

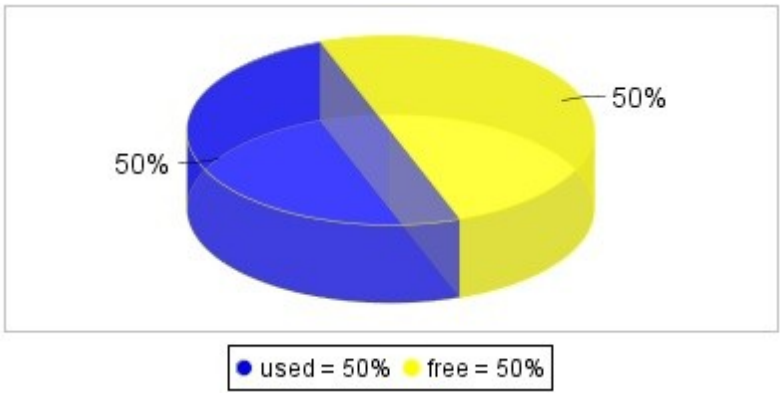


Graf č. 12 Zaznamenaný počet 231 odchozích SNMP zpráv směrovače Router1 během testování

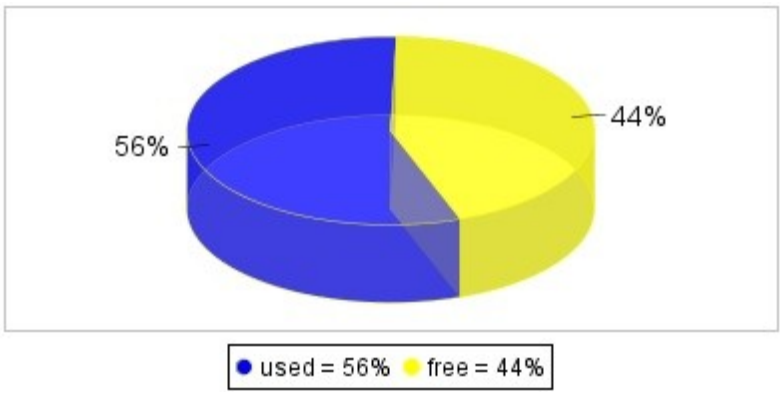
Příloha.D: Zaznamenané hodnoty zatížení CPU přepínače Switch1 během DoS útoku



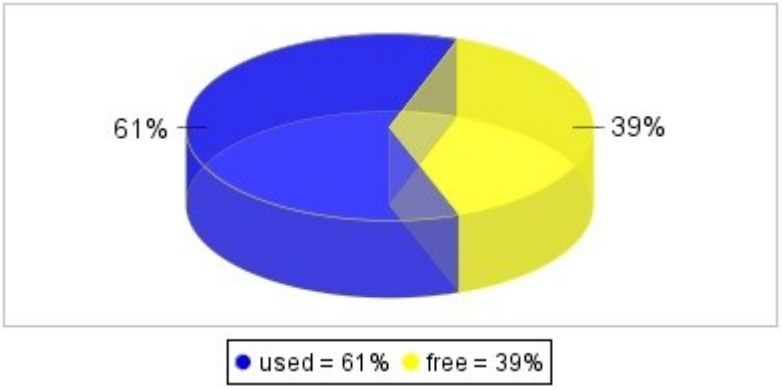
Graf č. 13 Využití CPU přepínače Switch1 po 1 minutě simulovaného DoS útoku



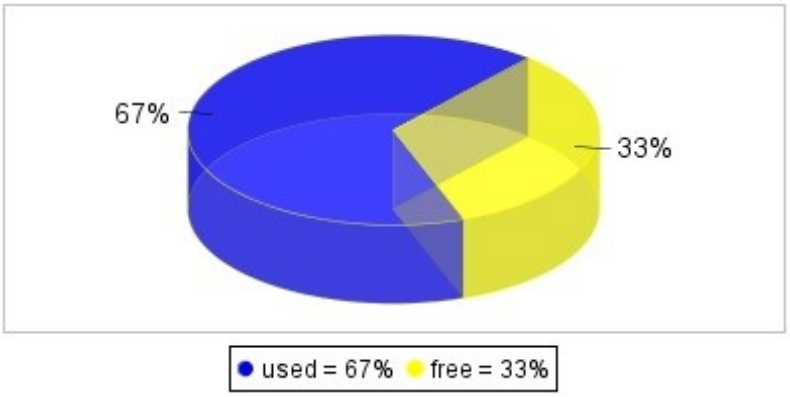
Graf č. 14 Využití CPU přepínače Switch1 po 2 minutách simulovaného DoS útoku



Graf č. 15 Využití CPU přepínače Switch1 po 3 minutách simulovaného DoS útoku



Graf č. 16 Využití CPU přepínače Switch1 po 4 minutách simulovaného DoS útoku



Graf č. 17 Využití CPU přepínače Switch1 po 5 minutách simulovaného DoS útoku

	atIfIndex	atPhysAddress	atNetAddress	Index Value
1	2	00-17-5A-4B-53-59	10.0.40.1	2.1.10.0.40.1
2	2	00-30-05-8E-5E-19	10.0.40.2	2.1.10.0.40.2

Obrázek 7.1 Tabulka adresového překladu směrovače Router1

	atIfIndex	atPhysAddress	atNetAddress	Index Value
1	10	00-21-1B-B7-A4-41	10.0.0.1	10.1.10.0.0.1
2	10	00-17-5A-4B-52-F3	10.0.0.10	10.1.10.0.0.10
3	10	3C-07-54-20-C5-24	10.0.0.100	10.1.10.0.0.100
4	11	00-21-1B-B7-A4-42	10.0.10.1	11.1.10.0.10.1
5	11	00-17-5A-4B-52-F3	10.0.10.10	11.1.10.0.10.10
6	12	00-30-05-AF-77-20	10.0.20.2	12.1.10.0.20.2
7	12	00-17-5A-4B-52-F3	10.0.20.10	12.1.10.0.20.10

Obrázek 7.2 Tabulka adresového překladu směrovače Router2

	atIfIndex	atPhysAddress	atNetAddress	Index Value
1	5	00-21-1B-B7-A4-41	10.0.0.1	5.1.10.0.0.1
2	5	3C-07-54-20-C5-24	10.0.0.100	5.1.10.0.0.100
3	10	00-21-1B-B7-A4-42	10.0.10.1	10.1.10.0.10.1
4	10	00-30-05-8E-5E-4F	10.0.10.2	10.1.10.0.10.2
5	20	00-21-1B-B7-A4-43	10.0.20.1	20.1.10.0.20.1
6	20	00-30-05-AF-77-20	10.0.20.2	20.1.10.0.20.2

7.3 Tabulka adresového překladu přepínače Switch1

	ipAdEntAddr	ipAdEntIfIndex	ipAdEntNetMask	ipAdEntBcastA...	Index Value
1	10.0.30.2	3	255.255.255.252	1	10.0.30.2
2	10.0.40.1	2	255.255.255.0	1	10.0.40.1

Obrázek 7.4 Tabulka IP adres směrovače Router1

	ipAdEntAddr	ipAdEntIfIndex	ipAdEntNetMask	ipAdEntBcastA...	Index Value
1	10.0.0.10	10	255.255.255.0	1	10.0.0.10
2	10.0.10.10	11	255.255.255.0	1	10.0.10.10
3	10.0.20.10	12	255.255.255.0	1	10.0.20.10
4	10.0.30.1	3	255.255.255.252	1	10.0.30.1

Obrázek 7.5 Tabulka IP adres směrovače Router2

	ipAdEntAddr	ipAdEntIfIndex	ipAdEntNetMask	ipAdEntBcastA...	Index Value
1	10.0.0.1	5	255.255.255.0	1	10.0.0.1
2	10.0.10.1	10	255.255.255.0	1	10.0.10.1
3	10.0.20.1	20	255.255.255.0	1	10.0.20.1

Obrázek 7.6 Tabulka IP adres přepínače Switch1

	1	2	3	4	5
ipRouteDest	10.0.0.0	10.0.10.0	10.0.20.0	10.0.30.0	10.0.40.0
ipRouteIfIndex	0	0	0	3	2
ipRouteMetric1	0	0	0	0	0
ipRouteMetric2	-1	-1	-1	-1	-1
ipRouteMetric3	-1	-1	-1	-1	-1
ipRouteMetric4	-1	-1	-1	-1	-1
ipRouteNextHop	10.0.30.1	10.0.30.1	10.0.30.1	10.0.30.2	10.0.40.1
ipRouteType	indirect	indirect	indirect	direct	direct
ipRouteProto	local	local	local	local	local
ipRouteAge	19	19	19	0	0

Obrázek 7.7 Směrovací tabulka směrovače Router1

	1	2	3	4	5
ipRouteDest	10.0.0.0	10.0.10.0	10.0.20.0	10.0.30.0	10.0.40.0
ipRouteIfIndex	10	11	12	3	0
ipRouteMetric1	0	0	0	0	0
ipRouteMetric2	-1	-1	-1	-1	-1
ipRouteMetric3	-1	-1	-1	-1	-1
ipRouteMetric4	-1	-1	-1	-1	-1
ipRouteNextHop	10.0.0.10	10.0.10.10	10.0.20.10	10.0.30.1	10.0.30.2
ipRouteType	direct	direct	direct	direct	indirect
ipRouteProto	local	local	local	local	local
ipRouteAge	0	0	0	0	9

Obrázek 7.8 Směrovací tabulka směrovače Router2